MOLECULAR DYNAMICS SIMULATION OF THE CARBON NANOTUBE – SUBSTRATE THERMAL INTERFACE RESISTANCE

A Thesis
Presented to
The Academic Faculty

Molecular Dynamics Simulation of the Carbon Nanotube – Substrate Thermal Interface Resistance

By

Daniel John Rogers

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Mechanical Engineering

Georgia Institute of Technology

December 2009

MOLECULAR DYNAMICS SIMULATION OF THE CARBON NANOTUBE – SUBSTRATE THERMAL INTERFACE RESISTANCE

Approved by:

Dr. Jianmin Qu, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Matthew Yao
Rockwell Collins, Inc.

Dr. Yogendra Joshi
School of Mechanical Engineering
Georgia Institute of Technology

Dr. C.P. Wong
School of Material Science
Georgia Institute of Technology

Date Approved: July 28, 2009

To Mom and Dad

ACKNOWLEDGEMENTS

I wish to acknowledge my mentor and advisor Dr. Jianmin Qu who gave me a wonderful opportunity to learn and contribute to nanoscale thermal transport theory and Rockwell Collins Corporation for their financial support. I would also like to thank Narasimhan Swaninathan for his introduction to molecular dynamics simulation as well the rest of my colleagues for their support of this research. Finally I wish to thank my father, Thomas, my mother, Margaret, and my sisters, Elizabeth and Rebecca for their continued faith in me and great generosity.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

R	Position vector
n_i	Integer set
a_i	Primitive vectors
k	Wavevector
b	Reciprocal lattice vector
M	Atomic mass
r_o	Equilibrium spacing
n	Atomic index
E	Total system energy
V_{ij}	Pairwise potential energy
r_{ij}	Pairwise atomic radius
u_{ij}	Displacement
i	Atomic index, Mode index
j	Atomic index
V_o	Equilibrium potential energy
E_o	Total equilibrium potential energy
F	Atomic force
M	Atomic mass
t	Time
A	Wave amplitude
x	Spatial coordinate
ω	Frequency

n	Atom number, Mode occupation number
v_g	Group velocity
D	Density of states
N	Number of modes, Number of atoms
k_b	Boltzmann's constant
T	Temperature
v	Atomic velocity
u	Thermal energy density
V	Volume
R	Gas constant
c_V	Specific heat capacity
$\hbar$	Planck's constant
θ_D	Debye temperature
ω_D	Debye frequency
α	Scaling factor, Transmission Coefficient
p	Polarization
σ_K	Kapitza conductance
c	Mode speed
A	Interface area

TIM	Thermal interface material
PCM	Phase-change material
CNT	Carbon nanotube
BCC	Body-centered cubic
FCC	Face-centered cubic
MD	Molecular dynamics
NEMD	Non-equilibrium molecular dynamics

SUMMARY

Thermal management is a key challenge to improving the performance of microelectronic devices. For many high performance applications, the thermal resistance between chip and heat sink may account for half of the total thermal budget. Chip-level heat dissipation is therefore a critical bottleneck to the development of advanced microelectronics with high junction temperatures. Recently aligned carbon nanotube arrays have been developed as possible next generation thermal interface materials to overcome this thermal limitation, however the thermal physics of these nanoscale interfaces remains unclear. In this thesis, the thermal interface resistance between a carbon nanotube and adjoining carbon, silicon, or copper substrate is investigated through non-equilibrium molecular dynamics simulation. Phonon transmission is calculated using a simplified form of the diffuse mismatch model with direct simulation of the phonon density of states. The results of theory and simulation are reported as a function of temperature in order to estimate the importance of anharmonicity and inelastic scattering. The results of this work provide a better understand of the mechanisms of thermal transport to assist future carbon nanotube thermal interface material research and development.

CHAPTER 1. INTRODUCTION

1.1 Motivation

Recent advances in microelectronics performance have led to increased junction temperatures and decreased device volumes. As a result of these higher power densities, thermal management has emerged as a key limiting factor to future device performance gains. This is particularly important for many defense-related applications, such as power amplifiers for RF communication devices seen in Figure 1-1, where the heat flux may reach 2000 W/cm^2 [1].

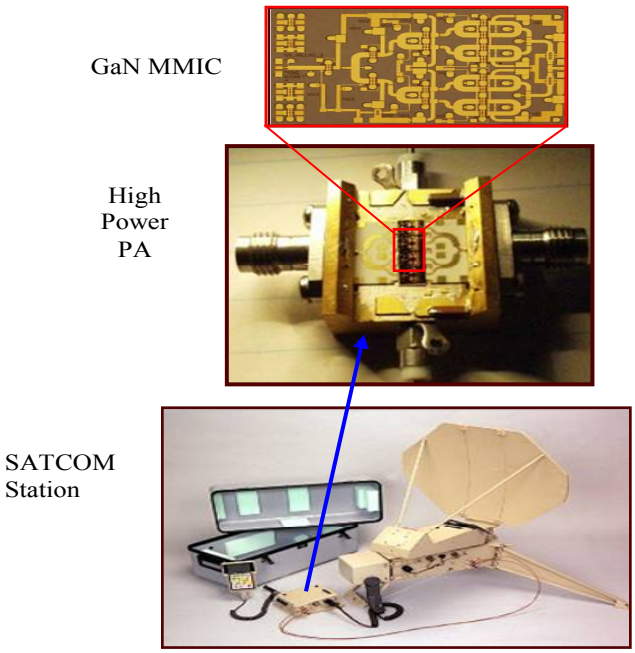

Figure 1-1. WBG Power Amplifier [1]

For example, with current thermal management technology the allowable power spike during image data transmission from a beyond-line-of-sight weapon in its terminal state is limited to 2.4 kbit/sec [1]. A 16x data rate improvement to 38.4 kbit/sec would greatly enhance the transferred image resolution as shown in Figure 1-2.

Figure 1-2. Thermal-limited image resolution vs. Ideal resolution [1]

Despite the clear performance improvements, high heat flux devices remain challenging to cool using even the best conventional thermal management techniques. For this reason, improvements in chip-level design and thermal interface material selection will have the greatest impact on thermally-limited, high-power chip performance.

1.2 Conventional Thermal Interface Materials

Thermal interface materials (TIMs) are used in microelectronics packaging to improve the thermo-mechanical properties of mesoscopic interfaces (Figure 1-3). Typical TIM materials include thermal greases, gels, phase change materials (PCMs), adhesives, solders, tapes, and pads [2]. TIM selection involves the optimization of numerous design considerations such as thermal conductivity, interface resistance, viscosity, temperature operating range, thermal cycling life, shock and vibration, shear strength, pressure, surface roughness, corrosion, ease of application, and cost [3]. Today many high-power, high-performance applications provide a real challenge to the current state-of-the-art TIM technologies where the chip-to-sink resistance may consume more

than half of the total thermal budget. Future microelectronics designs including three-dimensional circuits, new passivation layers, and new geometries (e.g. nanopillars and fin-shaped field-effect transistors) with increased metallic layers, current densities, and interconnect aspect ratios are projected to drive the chip-to-sink percentage of the thermal budget even higher. These trends in thermal management have led to a new interest in TIM thermal physics to complement the purely experimental history of TIM research [4].

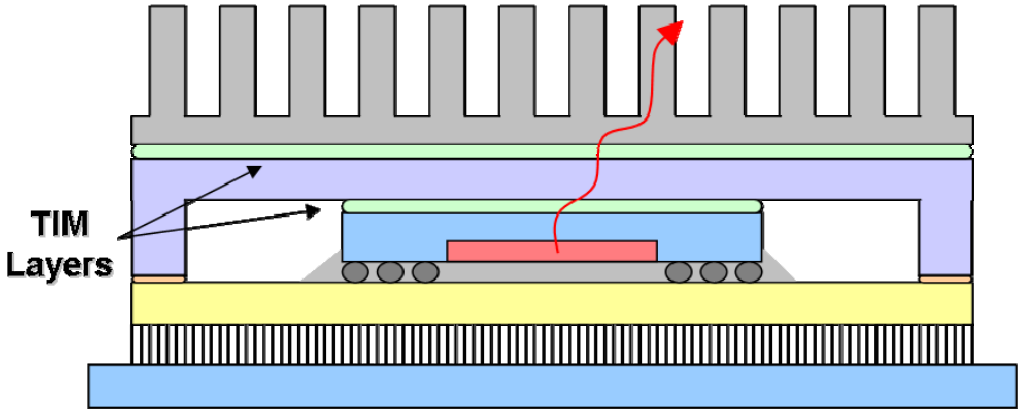

Figure 1-3. TIMs in Flipchip package

Thermal Greases

Thermal greases (Figure 1-4) are composed of a silicone base and ceramic or metallic filler. With increased temperature and pressure, thermal greases flow in between the heat sink and chip to fill the voids formed between the two rough surfaces. Greases have high thermal conductivity (3-5 W/mK), thermal stability, thin bond line thickness, low viscosity, and low cost [3-5]. Ceramic fillers are typically aluminum oxides or nitrides and metal fillers are silver or aluminum [3]. The disadvantages of thermal greases include uneven and cumbersome application, electrical contamination, 'pump-out,' and 'dry-out' [3, 4]. Electrical contamination may occur if the electrically

conductive grease shorts the electrical contacts with the chip. 'Pump-out' occurs when the thermal grease migrates due to the cyclic thermal expansion and contraction of the chip and sink materials.

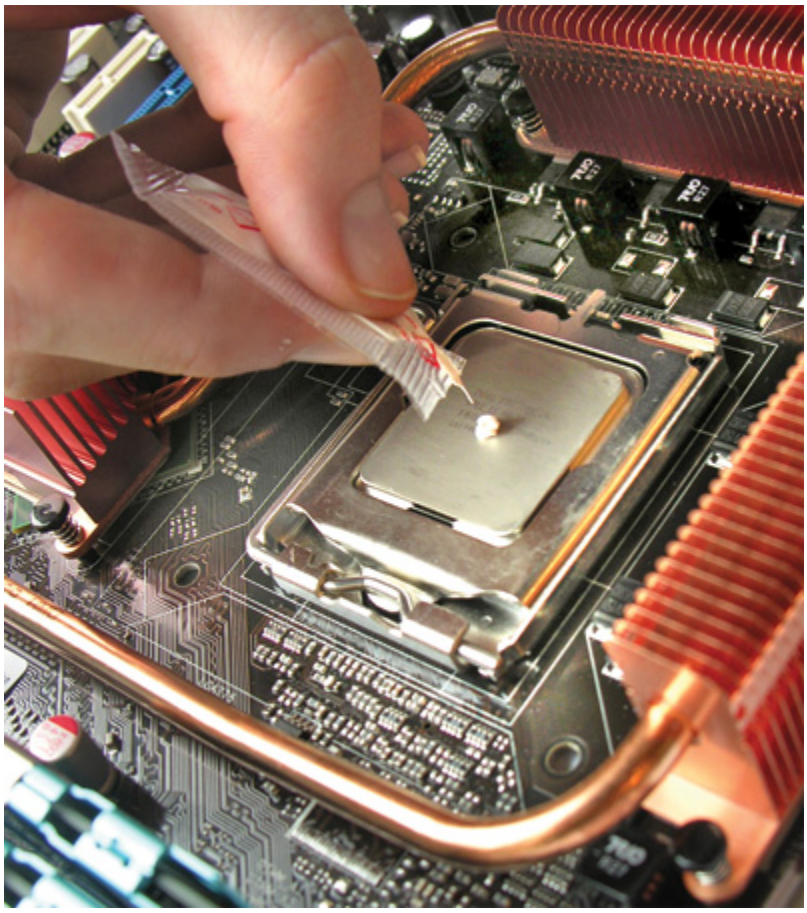

Figure 1-4. Application of Thermal Grease [6]

Phase-Change Materials

PCMs include polymeric pads with fillers and low-melting alloys [3, 4]. Thermal pads are solid at installation temperature but flow like greases at higher operating temperatures. PCM pads are easier to apply and are not susceptible to the migration, contamination, and dry-out problems of greases. The disadvantages of thermal pads include lower thermal conductivity (0.5-5 W/mK), higher interface resistance, and higher

attach pressure [3, 4]. Low-melting metal alloys, including bismuth, indium, gallium, and tin, allow for metal-metal thermal conduction across the interface but are susceptible to void formation, intermetallic growth, oxidation, and pump-out during thermal cycling [3].

Thermal Gels and Adhesives

Thermal gels are made of a silicone elastomer matrix filled with aluminum, silver, or ceramic fillers. Thermal gels function similarly to greases but have a curing step which eliminates migration but increases the application difficulty. Other disadvantages include delamination, lower thermal conductivity (3-4 W/mK), reduced flow into voids, and higher attach pressure [3, 4]. Thermal adhesives are epoxy-based with silver filler and have similar properties of gels except for a stiffer post-cure modulus resulting in additional mechanical stress at the interface [7].

Solders

Solders are advantageous due to their high thermal conductivity (30-50 W/mK), however their application is limited due to significant voiding, mechanical stiffness, intermetallic formation, thermal cycling life, increased cost, and difficult application [4, 8]. Voiding may be on the order of 24% [9].

Carbon-based Composites

Due to their high thermal conductivity, allotropes of carbon (carbon fibers, graphitic sheets or flakes, and fullerenes) have been researched as filler materials for TIM

composites [3, 10-14]. A carbon composite would ideally have the high thermal conductivity of a solder with the thermomechanical compliance of a polymer or gel. Unfortunately these composites have shown only modest improvement in the overall thermal conductivity, especially in comparison to the theoretically very high conductivity of the filler materials [12]. This marginal improvement suggests that the thermal transport is dominated by the numerous interfaces in the chip-to-sink pathway.

State-of-the-Art

Chung and Liu *et al*. compiled performance metrics for both commercial and experimental state-of-the-art TIMs [2, 8]. The best reported thermal interface resistance from Chung is approximately 5 mm^2K/W [8].

1.3 Aligned Carbon Nanotube Arrays

Recently aligned carbon nanotube array TIMs (Figure 1-5, Figure 1-6) have been developed in order to maximize thermal transport through the highly conductive carbon nanotubes (CNTs) in the direction perpendicular to the surface and to minimize the number of interfaces in the chip-to-sink thermal pathway [3]. The following paragraphs summarize the experimental results from the CNT TIM literature.

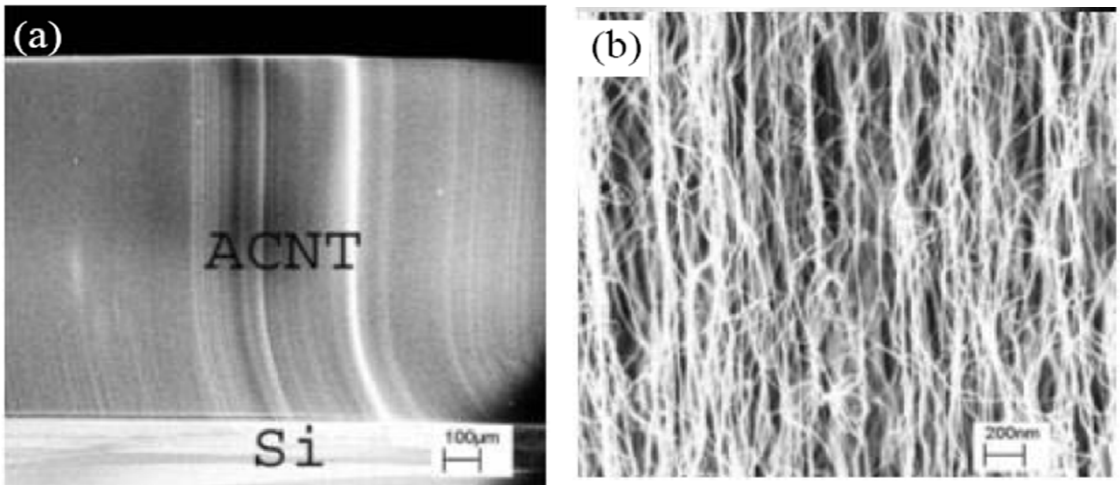

Figure 1-5. SEM image of aligned CNT array
Courtesy of Dr. C.P. Wong, Georgia Tech [15]

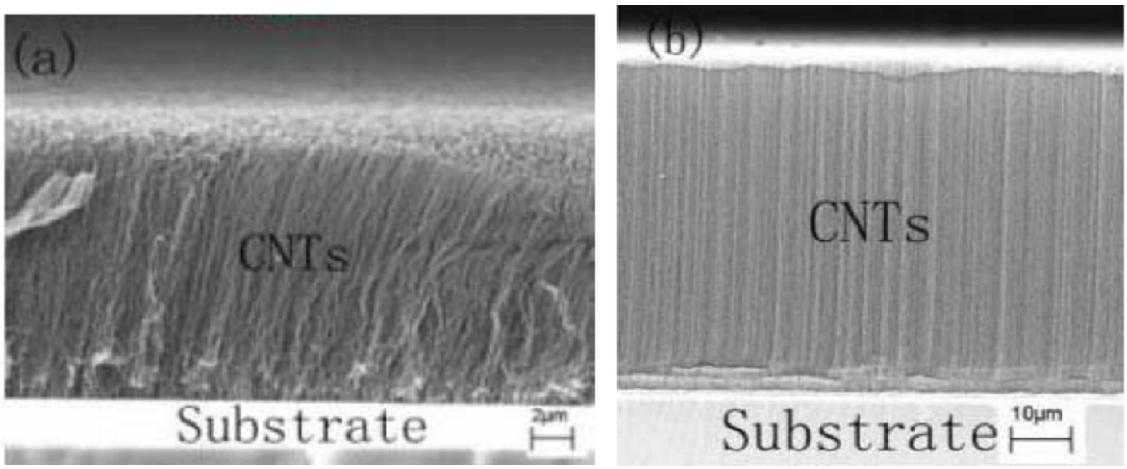

Figure 1-6. Aligned CNT array grown at (a) 650°C (b) 675°C
Courtesy of Dr. C.P. Wong, Georgia Tech [15]

Goodson *et al.* of Stanford University used a 3-omega MEMS-based frequency method to measure the thermal properties of an aligned CNT array and report CNT conductivity of 74 – 83 W/mK and interface resistance of ~15 mm^2K/W between 295 and 325 K with attached pressures 5.8 – 14.5 psi [16, 17]. To reduce the CNT-substrate interface resistance, the thermal properties of two opposing CNT arrays grown on silicon

and pressed together were measured using infrared microscopy [18]. Goodson *et al.* reported a large total resistance of 380 mm^2K/W for this double CNT array structure due to the very high CNT-CNT resistance [18]. The best CNT-TIM reported by Goodson *et al*. is a 28 μm CNT array grown on silicon with a 10 nm oxide layer and an iron catalyst [19]. A 20 nm palladium layer was first deposited onto the free end of the array, followed by a 160 nm aluminum layer. Using a transient thermoreflectance technique, the thermal properties reported are >8 W/mK CNT array conductivity, 2.9 mm^2K/W CNT-Pd interface resistance, and 9.1 mm^2K/W CNT-SiO_2 interface resistance [19].

Xu and Fisher of Purdue University first recorded a Cu-CNT-Cu thermal interface resistance of 23 mm^2K/W at 0.445 MPa for a ~7 μm CNT array using a one-dimensional reference bar method [20]. In a following studying, 10 μm CNT arrays were grown on silicon with 30 nm titanium, 10 nm aluminum, and 6 nm nickel catalyst layers [21]. The measured Cu-CNT-Si interface resistance improved to 19.8 mm^2K/W at 0.445 MPa with the new CNT array [21]. A phase change material yielding 16.2 mm^2K/W interface resistance at 0.35 MPa between copper and silicon improved to 5.2 mm^2K/W with the additional of the CNT array [21]. A third one-dimensional reference bar experiment by Xu and Fisher parameterized by growth method reported thermal interface resistances between 20 – 37 mm^2K/W at 0.445 MPa [22].

Since the one-dimensional reference bar method has limited accuracy on the order of 1 mm^2K/W, Cola *et al.* applied a photoacoustic technique to similarly grown 15 μm thick CNT array samples of Xu and Fisher [23]. Thermal interface resistances results are 1.7 ± 1.0 mm^2K/W for Si-CNT and 14.0 ± 0.9 mm^2K/W for Ag-CNT at 0.241 MPa [23]. For interfaces with CNT arrays grown on both sides, the interfaces resistances are $0.8 \pm$

0.5 mm^2K/W for Si-CNT, 2.1 ± 0.4 mm^2K/W for CNT-CNT, and 0.9 ± 0.5 mm^2K/W for Cu-CNT [23]. Cola *et al*. also tested for the effect of the multi-wall CNT diameter distribution with the array again using the photoacoustic technique [24]. For mean diameter distributions 23.2 – 70.3 nm and pressures 10 – 30 psi, the measured thermal interface resistances are between 8 – 16 mm^2K/W [24].

Double-sided CNT/foil structures were also tested using this photoacoustic technique and Cola *et al.* report a thermal interface resistance of ~10 mm^2K/W for these double array/foil structures [25]. Parameterization across growth temperatures yielded interface resistances between 7 – 19 mm^2K/W [26]. A SiC-CNT-Ag interface was investigated using the same photoacoustic technique and parameterized across interface temperatures [27]. Cola *et al*. reported results between 10-16 mm^2K/W with a slight decrease in interface resistance with increasing temperature [27].

Majumdar *et al.* of the University of California – Berkeley employed a phase sensitive transient thermo-reflectance technique to measure the thermal interface resistance of a glass-CNT-Si and glass-In-CNT-Si interface [28]. The CNT array is grown on 10 nm of aluminum and 10 nm of iron with a height of 7 μm. The reported thermal interface results for the glass-CNT-Si are 11 mm^2K/W and 1 mm^2K/W respectively [28]. The results for the In-CNT-Si are 0.29 mm^2K/W and 0.45 mm^2K/W, respectively [28].

Son *et al.* of Rensselaer Polytechnic Institute used a photothermoelectric technique and an analytical model to measure the thermal interface resistance of a Si/SiO_2– CNT array with zero pressure [29]. The experimental results show a 215 μm thick CNT array resistance of 48 mm^2K/W and 128 μm thick CNT array resistance of 49

mm^2K/W [29]. Using the theoretical model of Prasher *et al.* [30, 31], the theoretical prediction of thermal interface resistance is 0.44 mm^2K/W [29]. The difference between experimental results and theoretical predictions are attributed to imperfect contact between CNT and substrate due to the growth process and phonon-electron-phonon scattering at the metal catalyst interface [29].

Copper composite aligned CNT arrays have also been developed and tested using the reference bar method [32, 33]. Ngo *et al*. reported a Cu-CNT-Si thermal interface resistance of 25 mm^2K/W at 0.41 MPa for a 7.5 μm thick array [32]. Yang *et al.* reported a smaller resistance of 10 mm^2K/W at 0.1 MPa for a 40 percent copper composite [33]. Wu *et al.* developed a composite silicone rubber (PDMS) – CNT array with and without an evaporated aluminum layer between the array and silicon substrate [34]. Reference bar test results show a thermal interface resistance of 152 mm^2K/W without aluminum and 83 mm^2K/W with aluminum for a 0.15 mm thick sample [34]. Additional CNT array TIM were also developed Shaikh *et al*. [35, 36].

C.P. Wong *et al.* of the Georgia Institute of Technology have developed carbon nanotube arrays and pillars for electrical interconnects and thermal management [15, 37-41]. Caps on the multi-walled nanotube free ends are removed through water-assisted selective etching allowing for enhanced solder wetting (Figure 1-7) to improve electrical and thermal transport. Thermal radiation measurements of the soldered Cu-CNT structure result in a thermal resistance of 43 mm^2K/W for a 180 μm thick CNT array sample [42]. Functionalization through a 'chemical transfer' process has also been demonstrated as a means of covalently bonding the nanotubes to a substrate through self-assembled molecule layers [43, 44].

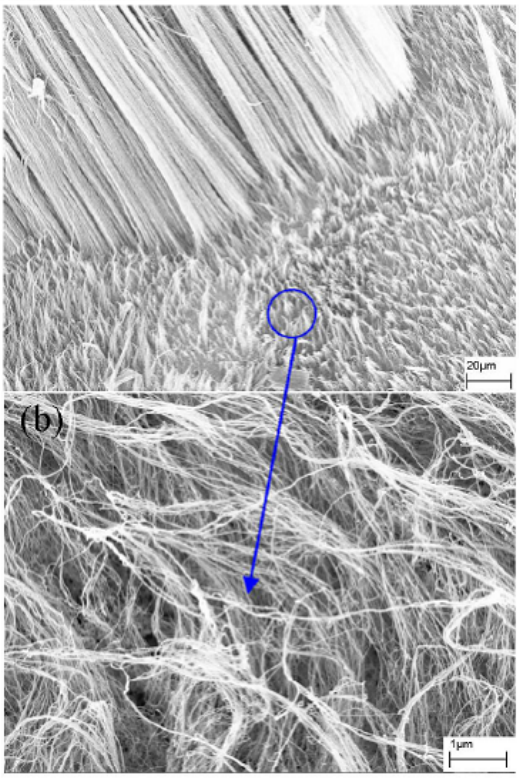

Figure 1-7. Soldered CNT array interface
Courtesy of C.P. Wong [42]

Kim *et al.* directly measured the thermal interface resistance of a multi-walled carbon nanotube suspended between two microfabricated silicon nitride pillars metalized with platinum functioning as a heater and thermometer [45]. The room temperature thermal conductivity was measured greater than 3000 W/mK and the interface resistance was measured as 0.028 mm^2K/W [45]. Goodson *et al.* compared the measured the heat capacity of a CNT TIM to an individual CNT to calculate an effective volume of CNTs involved in thermal transport [19]. This effective volume ratio was then used to estimate the individual platinum-CNT interface resistance of 0.010 mm^2K/W and silicon oxide-CNT interface resistance of 0.032 mm^2K/W [19].

In summary, current measurements of the thermal interface resistance of CNT array TIMs span a range ~0.1 – 1000 mm^2K/W. Large variations are due to differences in growth methods, array density, array quality, array thickness, catalyst layers, oxide layers, attach pressures, local temperatures, substrate materials, and measurement technique. Measurement of the small thermal resistance of a particular material interface in the overall TIM structure is an inherently challenging experimental task requiring use of non-ASTM standard techniques: 3-omega, thermoreflectance, photoacoustic, etc.

There is a fundamental difference in scale between conventional TIM materials and CNT TIMs. Conventional TIMs rely upon two mesoscopic material properties: thermal conductivity and mechanical compliance [28]. Greases, PCMs, adhesives, etc. fill the mesoscopic voids at the interface and have high bulk thermal conductivity through the addition of filler materials. For CNT TIMs, the thermal transport occurs through highly conductive CNTs functioning as one-dimensional phonon waveguides. The total thermal interface resistance is a sum of the phonon transport through each tube and interfacial phonon transmission at each tube point of contact. A full description of the thermal path of the CNT TIM structure therefore requires a multi-scale analysis beginning with each individual CNT – substrate thermal interface.

1.4 Overview

The objective of this thesis is to determine the nature of thermal transport across nanotube-substrates interface using molecular dynamics simulation. Substrate materials considered include silicon, diamond carbon, and copper. A non-equilibrium molecular dynamics method is used to calculate the classical-anharmonic thermal interface

resistance for each particular configuration. A frequency-based phonon transmission model is used to estimate the quasi-harmonic, quasi-classical thermal interface resistance and is compared to the molecular dynamics simulation. Thermal interface resistance results are reported as a function of temperature and interface disorder to estimate the role of lattice anharmonicity and the degree of inelastic phonon scattering at the interface.

Chapter 2 introduces the basic concepts of lattice dynamics through an elementary analysis of normal modes in a one-dimensional atomic chain. The wave space is discussed through a visual examination of the Brillouin zone. Phonons are introduced as the quantum analogue of classical normal modes. Chapter 3 introduces the molecular dynamics method. The relevant interatomic potentials, thermostatting techniques, and boundary conditions are reviewed. The non-equilibrium method is review as well as the calculation of phonon properties from molecular dynamics output. Chapter 4 defines the Kapitza conductance for the acoustic and diffuse mismatch models, lattice dynamical calculations, and molecular dynamics simulation in terms of phonon transmission. Chapter 5 contains the results of the molecular dynamics simulations and diffuse mismatch model calculations. Chapter 6 summarizes the results and conclusions. The Appendix contains all relevant original codes.

CHAPTER 2. LATTICE DYNAMICS

Many properties of solids such as heat conduction, thermal energy storage, thermal expansion, and melting result from the vibration of atoms. Unlike an ideal gas, atoms in a lattice do not vibrate randomly, but rather vibrate in an ordered fashion due to the interatomic forces of the physical and chemical bonds and the geometric symmetries of the lattice. In this chapter, Section 2.1 introduces the lattice structure in both the physical space and the reciprocal space. Section 2.2 and 2.3 apply these concepts to simple, one-dimensional, infinite, monatomic and diatomic lattices and derive the corresponding dispersion relations. Section 2.4 defines the density of states for the normal modes of a one-dimensional monatomic lattice. Section 2.5 reviews the models of specific heat capacity to relate classical normal modes to quantum phonons.

2.1 Crystal lattice structure

In real or physical space, the atomic positions of crystalline systems are defined according to the position vector, R

$$\vec{R} = n_1\vec{a}_1 + n_2\vec{a}_2 + n_3\vec{a}_3 \tag{2.1}$$

where a_i are the primitive vectors which span the basis of the lattice and n_i are a set of integers locating the atom site as seen in Figure 2-1 [46].

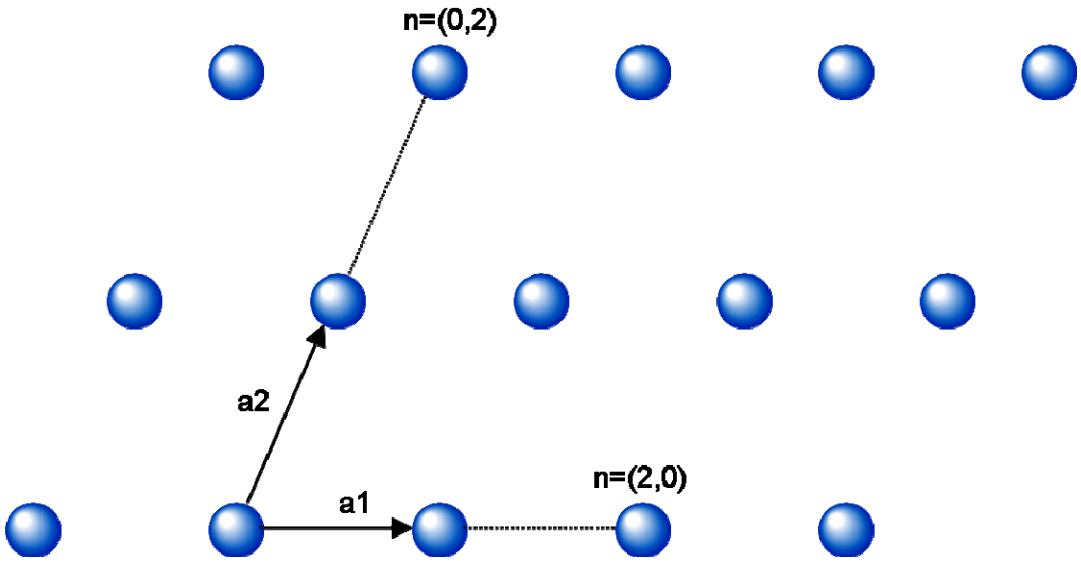

Figure 2-1. 2D Bravais lattice

This lattice type is known as a Bravais lattice or a lattice with a basis [47]. A primitive cell is defined using the primitive vectors of the Bravais lattice. The primitive cell nearest to a lattice point as shown in Figure 2-2 is referred to as a Wigner-Seitz cell [47].

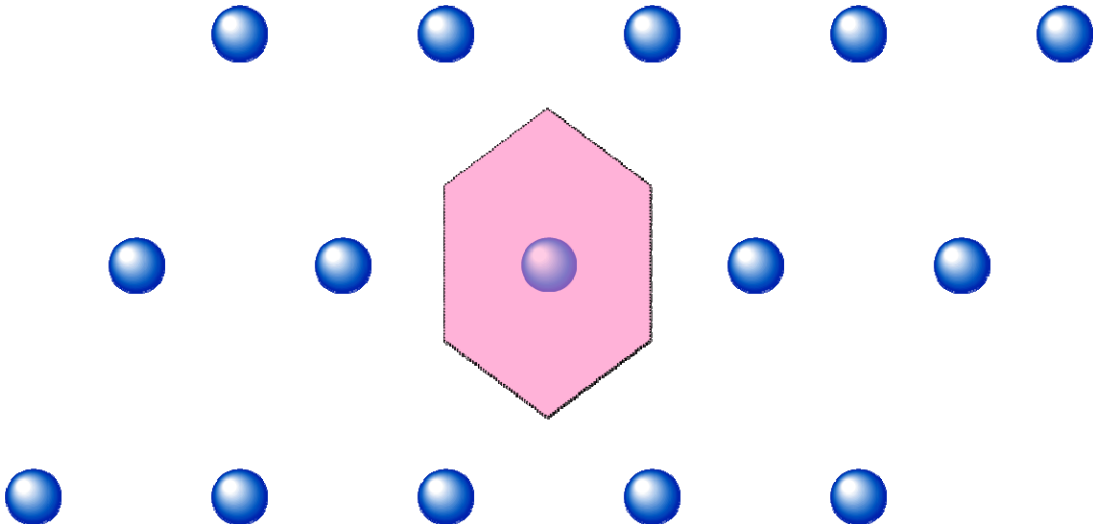

Figure 2-2. 2D Wigner-Seitz Cell

For wave analysis it is useful to observe the reciprocal space or wave space defined as the set of wavevectors, *k* such that [46]

$$e^{i\vec{k}\cdot\vec{R}} = 1 \tag{2.2}$$

A given wave length in real space is equivalent to a wavevector in reciprocal space.

Using the relation in Eq. (2.2), the reciprocal lattice vectors, *b* may be defined in terms of the primitive vectors, *a* as [46]

$$\begin{aligned}
\vec{b}_1 &= 2\pi \frac{\vec{a}_2 \times \vec{a}_3}{\vec{a}_1 \cdot (\vec{a}_2 \times \vec{a}_3)} \\
\vec{b}_2 &= 2\pi \frac{\vec{a}_3 \times \vec{a}_1}{\vec{a}_1 \cdot (\vec{a}_2 \times \vec{a}_3)} \\
\vec{b}_3 &= 2\pi \frac{\vec{a}_1 \times \vec{a}_2}{\vec{a}_1 \cdot (\vec{a}_2 \times \vec{a}_3)}
\end{aligned} \tag{2.3}$$

A Wigner-Seitz cell containing the fundamental wavevectors in reciprocal space is known as the first Brillouin zone [47]. In Figure 2-3 the Wigner-Seitz cells are shown for three-dimensional body-centered cubic (BCC) and face-centered cubic (FCC) crystal structures in both the real and reciprocal spaces. Note that the BCC Brillouin zone in reciprocal space is equivalent to the FCC Wigner-Seitz cell in real space.

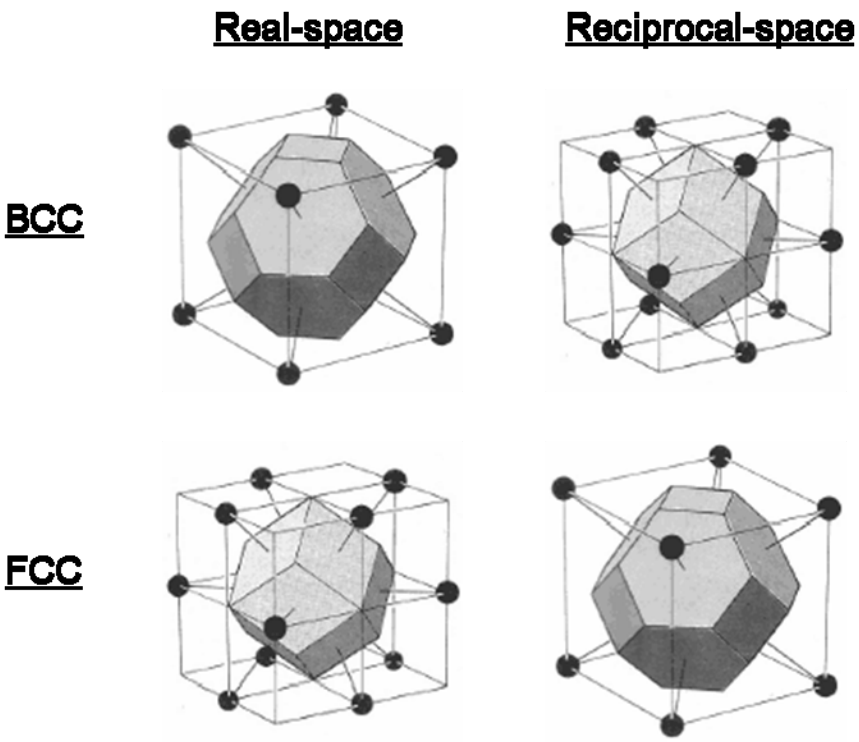

Figure 2-3. 3D Wigner-Seitz and Brillouin Zones [48]

The surfaces of the Wigner-Seitz cells in reciprocal space form planes known as Bragg planes [47]. Higher order Brillouin zones are defined by number of number of Bragg plane intersections they contain [47]. The second Brillouin zone is the set of points reached from the first Brillouin zone by crossing only one Bragg plane [47]. The *n*-th Brillouin zone is the set of points reached by crossing *n*-1 Bragg planes. The first, second, and third Brillouin zones and Bragg planes are shown for a two-dimensional square lattice in Figure 2-4. The first Brillouin zone is often simply referred to as the Brillouin zone in lattice dynamics since the first Brillouin zone contains all of the primitive wavevectors which are sufficient to completely describe the wave motion. The degeneracy of wavevectors in higher order Brillouin zones in discussed in Section 2.2 (Figure 2-7).

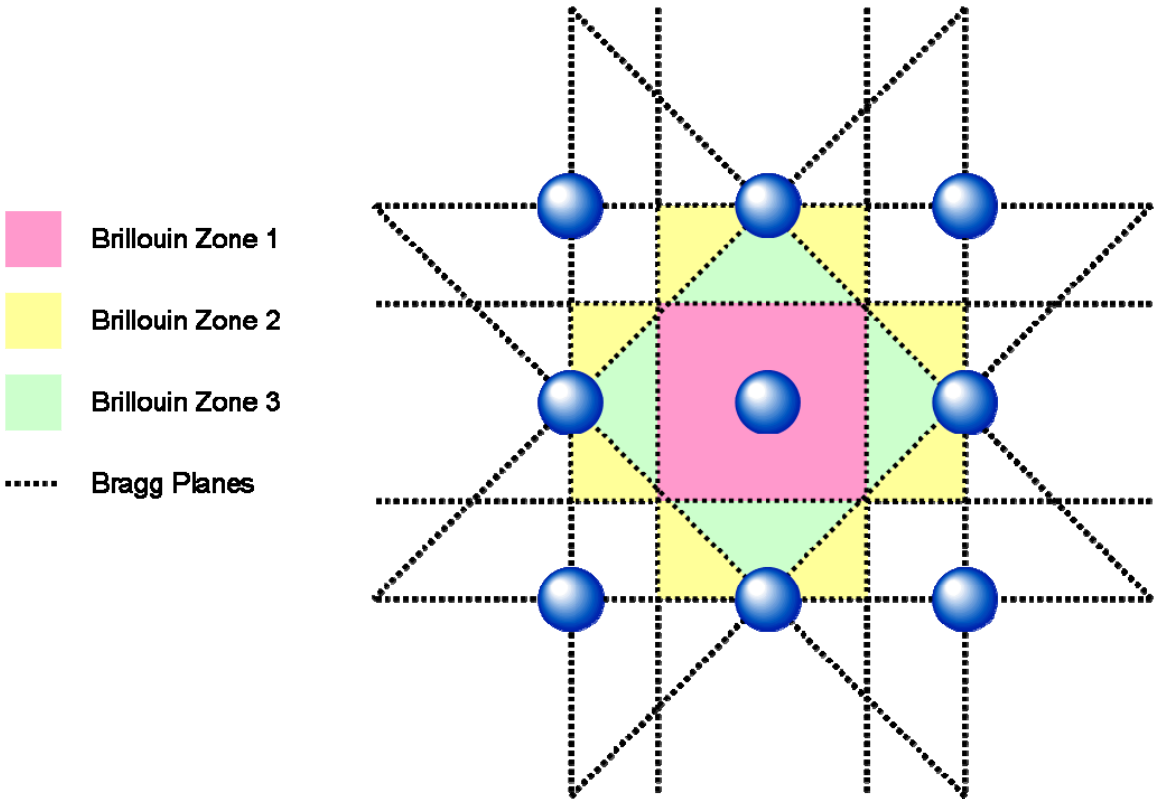

Figure 2-4. 2D Square Lattice Brillouin Zones and Bragg Planes

2.2 Monatomic Lattice

The one-dimensional chain of atoms shown in Figure 2-5 represents an infinitely long monatomic lattice with atomic mass, M equilibrium spacing, r_o and atomic index, n = 1,2,3,…

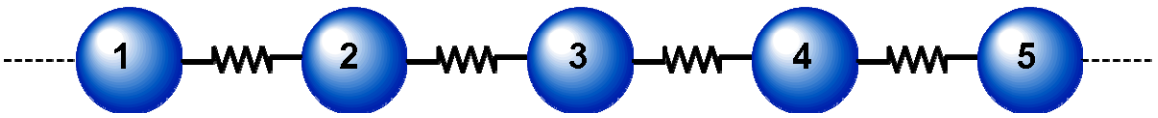

Figure 2-5. One dimensional monatomic chain

The total energy of this system, E is expressed as the sum of the potential energy of each atom pair

$$V_{ij} = V(r_{ij}) \tag{2.4}$$

$$E_{total} = \sum_{i} \sum_{j>0} V(r_{ij}) \tag{2.5}$$

where $V(r_{ij})$ is the pairwise potential energy and r_{ij} is the instantaneous pairwise spacing expressed as a function of displacement of each atom, u_{ij} and the equilibrium spacing, r_o

$$r_{ij} = r_o + u_j - u_i \tag{2.6}$$

Expanding the expression for pairwise potential energy about the equilibrium spacing using a Taylor series,

$$V_{ij} = V_o + V_o'(u_j - u_i) + \frac{1}{2} V_o''(u_j - u_i)^2 + \ldots \tag{2.7}$$

where V_o is the potential energy of the equilibrium spacing. Assuming near-neighbor interaction, minimum potential energy at equilibrium, and harmonicity, the following conditions apply respectively:

$$\begin{aligned} j &= i + 1 \\ V' &= 0 \\ V^n &= 0; n = 3,4,5,\ldots \end{aligned} \tag{2.8}$$

The simplified pairwise potential energy and total energy are now expressed as

$$\begin{aligned} V_{ij} &= V_o + \frac{1}{2} V_o''(u_{i+1} - u_i)^2 \\ E_{total} &= E_o + \frac{1}{2} V_o'' \sum_i (u_{i+1} - u_i)^2 \end{aligned} \tag{2.9}$$

With an expression for the total potential energy as a function of each atomic displacement, the force on a particular atom, alpha, may be expressed as

$$F_\alpha = -\frac{\partial E_{total}}{\partial u_\alpha} = V_o''(u_{\alpha+1} - 2u_\alpha + u_{\alpha-1}) \tag{2.10}$$

and applying Newton's second law,

$$M\frac{d^2u_\alpha}{dt^2} = V_o''(u_{\alpha+1} - 2u_\alpha + u_{\alpha-1}) \tag{2.11}$$

The solution to the equation of motion (2.11) is the wave equation

$$u_\alpha = Ae^{i(kx-\omega t)} \tag{2.12}$$

where A is the amplitude of displacement, k is the wavevector, x is the spatial coordinate, ω is the frequency, and t is the time. The dispersion relation is calculated by substituting the wave equation (2.12) into the equation of motion (2.11). First the displacement is calculated for each atom adjacent to atom, α.

$$\begin{aligned} x &= nr_o \\ u_\alpha &= Ae^{-i\omega t}e^{iknr_o} \\ u_{\alpha+1} &= Ae^{-i\omega t}e^{iknr_o}e^{ikr_o} \\ u_{\alpha-1} &= Ae^{-i\omega t}e^{iknr_o}e^{-ikr_o} \end{aligned} \tag{2.13}$$

Next the displacement equations are substituted into the equation of motion (2.11) and simplified as

$$
\begin{aligned}
-M\omega^2 u_\alpha &= V_o''(u_\alpha e^{ikr_o} + u_\alpha e^{-ikr_o} - 2u_\alpha) \\
-M\omega^2 &= 2V_o''(\cos kr_o - 1) \\
M\omega^2 &= 4V_o'' \sin^2(\frac{kr_o}{2})
\end{aligned}
\tag{2.14}
$$

Dividing by the mass and taking the square root, the frequency can be expressed as a function of the wavevector

$$\omega = 2\sqrt{\frac{V_o''}{M}}\left|\sin(\frac{kr_o}{2})\right| \tag{2.15}$$

Equation (2.15) is known as the dispersion relation, i.e. the relationship between frequency (time) and wavevector (space), for a one-dimensional, monatomic, harmonic lattice. The group velocity is

$$v_g = \frac{\partial\omega}{\partial k} = r_o\sqrt{\frac{V_o''}{M}}\left|\cos(\frac{kr_o}{2})\right| \tag{2.16}$$

For spring constant, V_o'' = 52.9 ev/Å^2, atomic mass, M = 12.0107 amu, and equilibrium spacing, r_o = 1.39 Å, the dispersion in Equation (2.15) is plotted in Figure 2-6 for the first and second Brillouin zones.

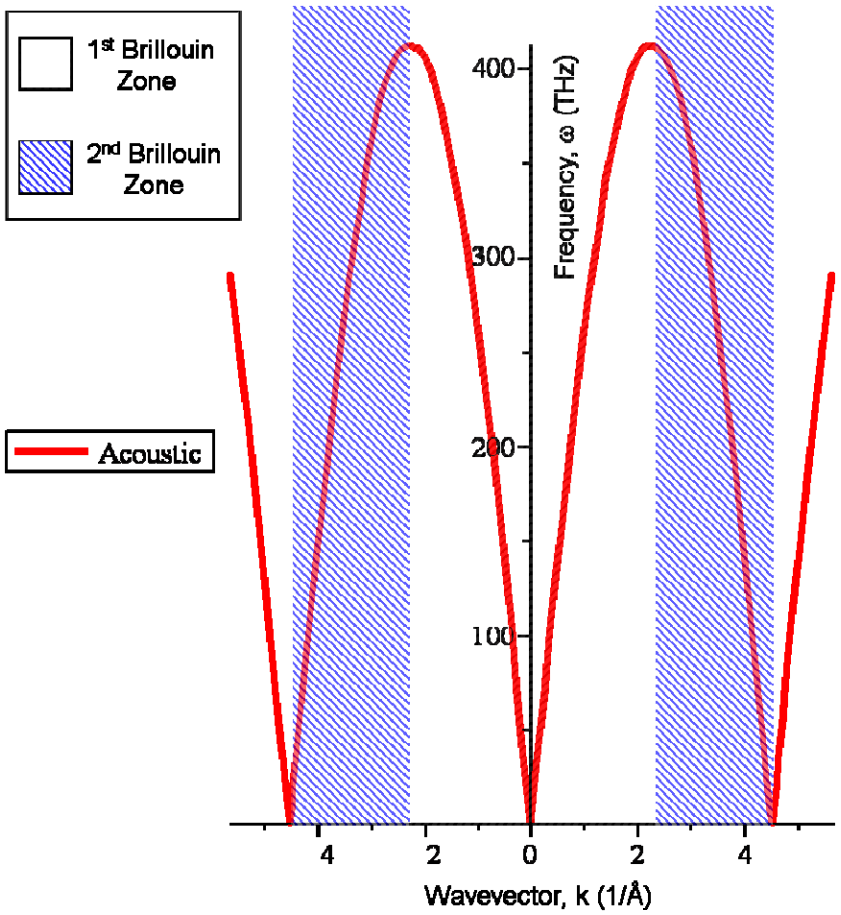

Figure 2-6. Dispersion Curve and Brillouin Zones for Monatomic Lattice

As mentioned in the previous section, the first Brillouin zone is the set of primitive wavevectors which fully describe the atomic motion. The difference between Brillouin zones is demonstrated by selecting wavevectors of the dispersion relation from three different Brillouin zones all with the same frequency. The resultant displacement is calculated for each atom in a monatomic chain using Eq. (2.15) and Eq. (2.12). Figure 2-7 illustrates that though the wavevectors differ, the atomic displacement is the same for each Brillouin zone. The first Brillouin zone therefore contains the primitive wavevectors necessary to describe all wave motion because higher order wavevectors in higher order Brillouin zones are degenerate to these primitive wavevectors.

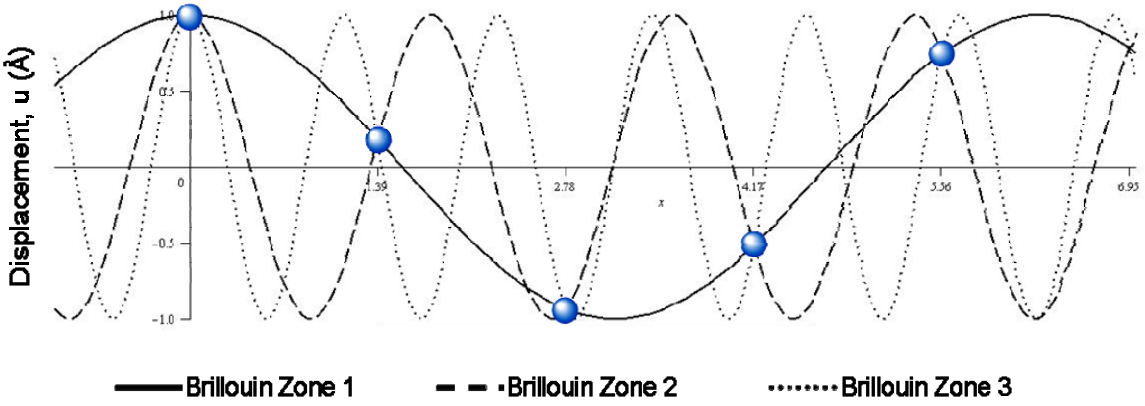

Figure 2-7. Brillouin Zone Degeneracy

2.3 Diatomic Lattice

Now consider a diatomic, infinite, one-dimensional lattice shown in Figure 2-8

Figure 2-8. Diatomic Lattice

consisting of two different atomic masses, M and m, all evenly spaced at equilibrium, r_o. There are now two equations of motion for the diatomic lattice

$$M\frac{d^2u_\alpha}{dt^2} = V_o''(u_{\alpha+1} - 2u_\alpha + u_{\alpha-1})$$
$$m\frac{d^2u_{\alpha+1}}{dt^2} = V_o''(u_{\alpha+2} - 2u_{\alpha+1} + u_\alpha) \qquad (2.17)$$

and two wave equations,

$$u_\alpha = Ae^{i(knr_o - \omega t)}$$
$$u_{\alpha+1} = Ae^{i(k(n+1)r_o - \omega t)} \tag{2.18}$$

the combination of which yields two homogenous equations for A and B.

$$A(M\omega^2 - 2V'') + B(2V''\cos kr_o) = 0$$
$$A(2V''\cos kr_o) + B(m\omega^2 - 2V'') = 0 \tag{2.19}$$

The nontrivial solution is given by the determinant of

$$\begin{vmatrix} M\omega^2 - 2V'' & 2V''\cos kr_o \\ 2V''\cos kr_o & m\omega^2 - 2V'' \end{vmatrix} = 0 \tag{2.20}$$

which when expanded yields the phonon dispersion curves for the diatomic lattice in Eq. (2.22).

$$(M\omega^2 - 2V'')(m\omega^2 - 2V'') - 4V''^2\cos^2 kr_o = 0 \tag{2.21}$$

$$\omega = \sqrt{V''(\frac{1}{M} + \frac{1}{m}) \pm V''\sqrt{(\frac{1}{M} + \frac{1}{m})^2 - \frac{4\sin^2 kr_o}{mM}}} \tag{2.22}$$

There are now two branches, one optical and one acoustic, of the dispersion curve for the diatomic lattice as displayed in Figure 2-9. The fundamental modes of vibration or normal modes are therefore identified by a frequency (time), a wavevector (space), and a branch (acoustic or optical). The acoustic branch sharply approaches zero frequency at zero wavevector meaning the long wavelength acoustic modes have low frequency and high group velocity. For the optical branch, as the wavevector approaches zero, the

frequency remains non-zero and the slope approaches zero. The optical modes are therefore higher frequency, near-standing waves at long wavelengths.

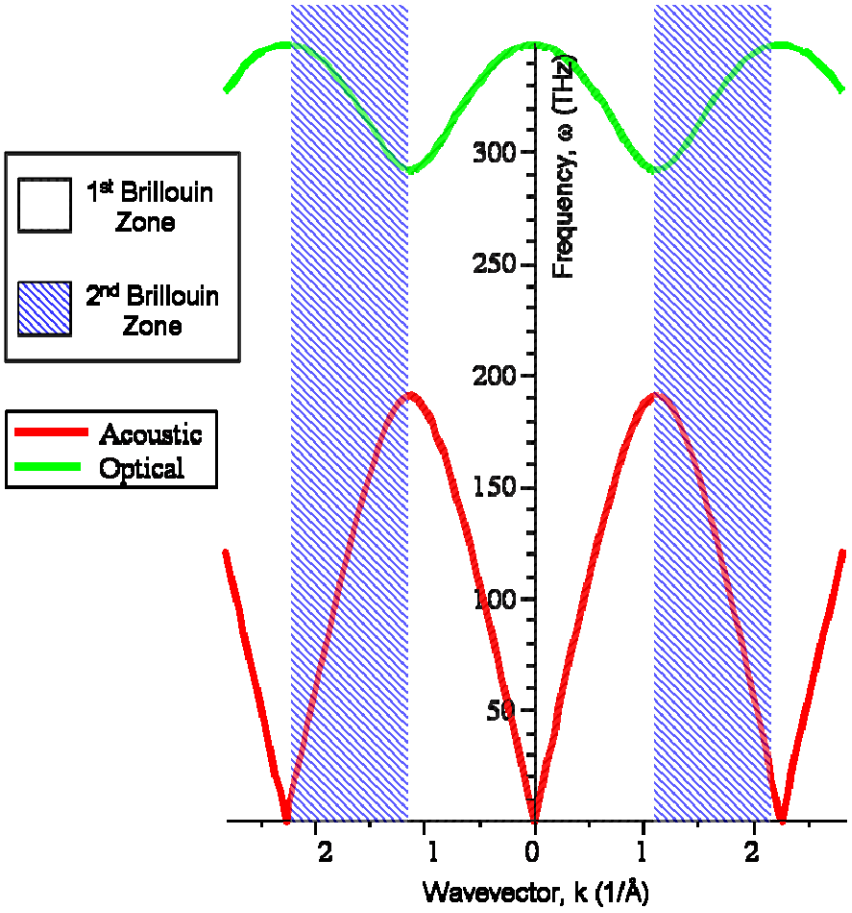

Figure 2-9. Acoustic & Optical Mode Dispersion Curves and Brillouin Zones

For two and three-dimensional lattices, there is a fourth characteristic of a normal mode known as the polarization. The polarization refers to the direction of atomic displacement relative to the direction of the wave motion. If the vector of atomic displacement is parallel to the wavevector, then the polarization is referred to as longitudinal. If the displacement vector is orthogonal to the wavevector, the polarization is transverse. In summary, a normal mode is defined by a frequency (time), wavevector (space), branch (acoustic or optical), and polarization (longitudinal or transverse).

2.4 Density of States

For a finite lattice of N atoms, there are a discrete number of modes vibrating at each frequency. The number of modes per unit volume as a function of frequency is known as the density of modes or density of states (DOS). For the monatomic lattice in Figure 2-5, the total number of modes per unit wavevector is

$$D'(\omega) = \frac{N}{\pi / r_o} \tag{2.23}$$

where $D'(\omega)$ is the DOS in wavevector space and N is the number of atoms. Since the number of modes in subset of wavevector space must be equal to the number of modes in the corresponding subset of frequency space, the DOS in frequency space may be written as

$$D(\omega) = D'(\omega) \cdot \frac{\partial k}{\partial \omega} \tag{2.24}$$

Applying the dispersion relation Eq. (2.15), the DOS for the finite, monatomic, one-dimensional lattice is

$$D(\omega) = 2\frac{N}{\pi}\sqrt{\frac{M}{V_o''}}\left(4 - \frac{M\omega^2}{V_o''}\right)^{-1/2} \tag{2.25}$$

and is plotted by frequency for the previous parameters in Figure 2-10.

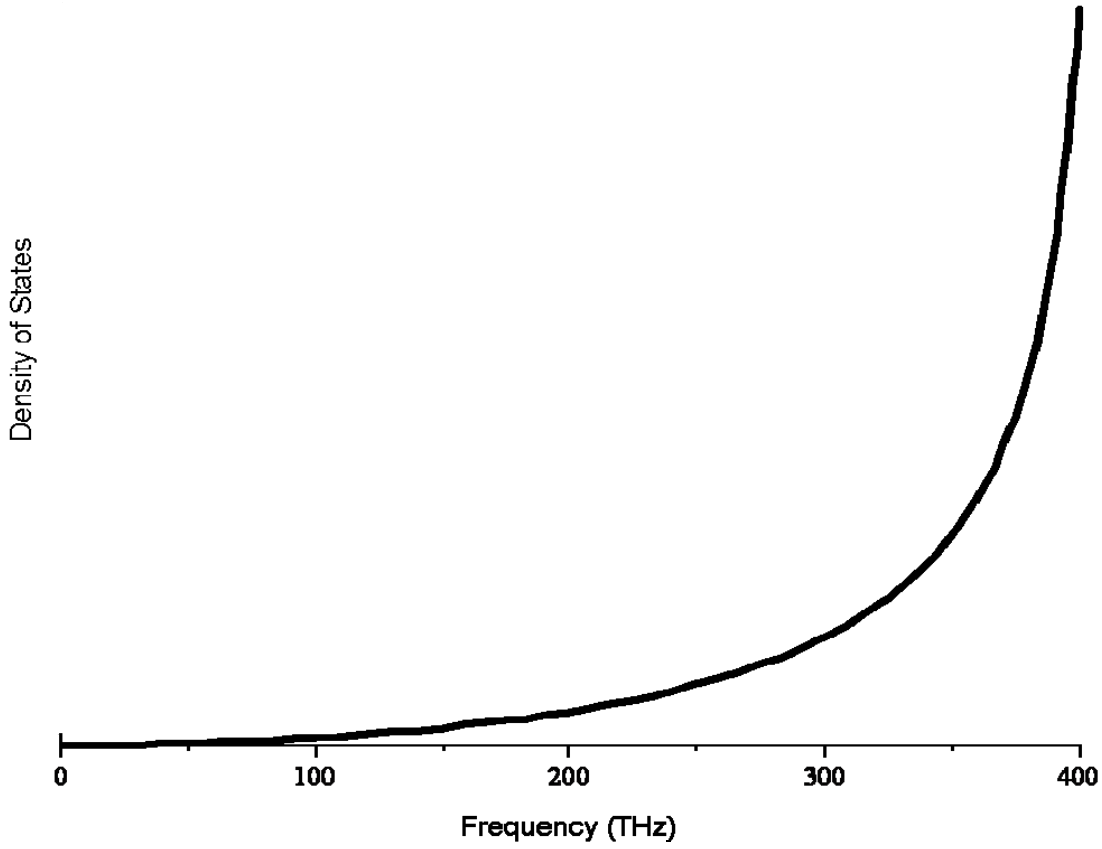

Figure 2-10. Density of States for One-Dimensional, Monatomic Chain

2.5 Classical Normal Mode vs. Quantum Phonon

In 1819 Dulong and Petit first observed that the specific heat capacity of a crystal was approximately three times the gas constant per unit molar mass [46]. In 1876 Boltzmann provided a theoretical basis for this observation with the equipartition theorem which states that the thermal energy is on average divided equally among each degree of freedom. The equipartition theorem may be expressed as

$$\frac{3}{2}Nk_bT = \sum_i \frac{1}{2}m_i v_i^2 \tag{2.26}$$

where N is the number of atoms, k_b is Boltzmann's constant, T is the temperature, m is the atomic mass, and v is the atomic velocity.

The Dulong-Petit law provided an accurate description of specific heat at high temperatures, but diverged significantly from experimental results at lower temperatures. This discrepancy led to Einstein's development of the quantum theory of crystals. In Einstein's theory, each atom is treated as an independent quantum-harmonic oscillator [46]. The energy or amplitude of vibration is not continuous with temperature as in the Boltzmann description, but instead is discrete with frequency [46]. In 1912 Debye refined the Einstein model by describing the coupled vibrations of atoms as quantum-harmonic waves with a spectrum of frequencies [49]. The specific heat models of Dulong-Petit, Einstein, and Debye are shown in Figure 2-11.

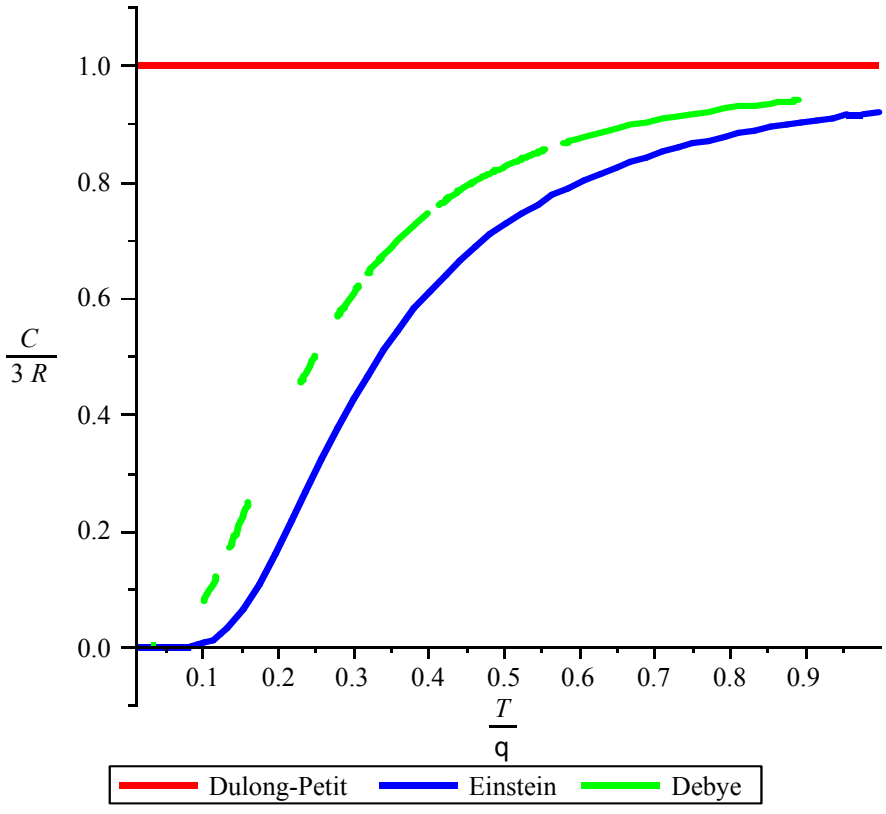

Figure 2-11. Comparison of Specific Heat Capacity Models

The discrepancy between the Dulong-Petit model and the Debye model for specific heat capacity at low temperatures is due to the difference in the treatment of energy in classical mechanics versus quantum mechanics. In classical mechanics, wave motion is due to the ordered oscillation of atoms in a lattice in response to classical interatomic forces resulting in normal modes [46, 47, 49]. The specific heat capacity, c_V derived from the equipartition of energy among each degree of freedom according to Maxwell-Boltzmann statistics [46, 47, 49] as seen in Eq. (2.27)

$$\begin{aligned} u &= u_o + 3nk_bT \\ c_V &= \left(\frac{du}{dT}\right)_V = 3nk_b = 3R \end{aligned} \tag{2.27}$$

where u is the thermal energy density, n is the number of atoms, k_b is Boltzmann's constant, T is the temperature, V is the volume, and R is the gas constant [47].

In quantum mechanics, the energy of the normal modes is quantized into elementary excitations known as phonons [46, 47, 49]. These phonons are quasi-particles with discrete packets of energy (ħω) and obey Bose-Einstein statistics [46, 47, 49]. The specific heat capacity of a quantum system of phonons is

$$\begin{aligned} u &= u_o + \frac{1}{V}\sum_i \hbar\omega_i\left(n_i + \frac{1}{2}\right) \\ n_i &= \frac{1}{e^{\hbar\omega/k_bT} - 1} \\ c_V &= \left(\frac{du}{dT}\right)_V = \frac{1}{V}\sum_i \frac{\partial}{\partial T}\left(\frac{\hbar\omega_i}{e^{\hbar\omega_i/k_bT} - 1}\right) \end{aligned} \tag{2.28}$$

where h is Planck's constant, ω_i is the mode frequency, n_i is the mode occupation number, and i is the phonon mode index for the summation over all phonon modes [47].

The temperature above which the Debye quantum model is equivalent to the classical Dulong-Petit model of specific heat capacity (Figure 2-11) is referred to as the Debye temperature [47]. Below the Debye temperature, higher frequency phonon modes are "frozen" out due to the energy quantization [47]. In other words, the Debye temperature corresponds to the minimum level of thermal energy necessary to excite the individual phonon mode with the largest quantum of energy or highest frequency ($\hbar\omega$). This maximum frequency is the Debye frequency and is related to the Debye temperature according to

$$k_b\theta_D = \hbar\omega_D \tag{2.29}$$

where θ_D is the Debye temperature and ω_D is the Debye frequency.

CHAPTER 3. MOLECULAR DYNAMICS SIMULATION

Molecular dynamics (MD) is a computational method that numerically solves Newton's equations for large systems classical particles in a statistical ensemble. Each particle or atom is considered as a point mass and interacts with other atoms through user-defined interatomic potentials. Since MD simulations are neither theory nor experiment, they are considered a "numerical experiment" where the relevant information of an ensemble of atoms or molecules may be derived from a limited input of masses and forces [50]. The advantage of MD is that it allows for investigation of complex conditions where theoretical approximations are insufficiently accurate and experiments are impractical. MD simulations have been applied to solid mechanics, fluid dynamics, gas dynamics, bio-molecules, and polymers to determine diffusion and transport properties, physical constants, size effects, etc. [50]. This chapter reviews the interatomic potentials, integrators, thermostats, non-equilibrium methods, phonon calculations, and other MD topics related to the nanotube-substrate thermal interface.

3.1 Interatomic Potentials

Silicon

MD simulation has been used to measure the thermal conductivity and thermal interface resistance of bulk silicon, silicon films, and nano-crystalline silicon [51-66]. The most common interatomic potential is the Stillinger-Weber potential [67] and other potentials include the Tersoff potential [68], a modified embedded-atom method (MEAM) [69], BKS potential [70], polymer consistent force field (PCFF) [71],

environment-dependent interatomic potential (EDIP) [72], and the extended Brenner potential (XB) [73-75].

Carbon

The thermal conductivity of carbon nanotubes has been modeled using molecular dynamics by Lukes and Zhong, Maruyama and Shigeo, and others [76-88]. The interatomic potentials used are the PCFF, the universal force field (UFF) [89], the reactive empirical bond order (REBO) potential [90], the Brenner potential [91] in simplified form [92], and the Tersoff potential [93-95]. Carbon nanotube thermal interfaces have also been considered [96-99], as well as the thermal conductivity of polyethylene chains [100], diamond carbon [101], and silicon carbide [102].

Copper

MD simulation has been used to estimate the phonon contribution to the thermal conductivity [103, 104] and thermal interface resistance [105] using the embedded-atom method (EAM) [106].

Brenner Potential

The primary interatomic potential used in this thesis is the Brenner or Tersoff-Brenner potential [91] in simplified form [92]. The Brenner potential is selected for its ability to accurately produce the phonon dispersion relations and density of states for carbon nanotube and silicon systems [73-75, 82]. The Brenner potential is a classical many-body potential written similarly to Eq. (2.5) as

$$
\begin{aligned}
E_{total} &= \sum_i \sum_{j>i} V_R(r_{ij}) - \bar{B}_{ij} V_A(r_{ij}) \\
V_R(r_{ij}) &= \frac{f_{ij}(r_{ij}) D_{ij}}{(S_{ij}-1)} e^{-\sqrt{2S_{ij}}\beta_{ij}(r-R_{ij})} \\
V_A(r_{ij}) &= \frac{f_{ij}(r_{ij}) D_{ij} S_{ij}}{(S_{ij}-1)} e^{-\sqrt{2/S_{ij}}\beta_{ij}(r-R_{ij})} \\
f_{ij} &= \begin{cases} 1 & r < R_{ij}^{(1)} \\ \frac{1}{2} + \frac{1}{2}\cos\left(\frac{\pi(r-R_{ij}^{(1)})}{R_{ij}^{(2)} - R_{ij}^{(1)}}\right) & R_{ij}^{(1)} < r < R_{ij}^{(2)} \\ 0 & r > R_{ij}^{(2)} \end{cases} \\
\bar{B}_{ij} &= \frac{B_{ij} + B_{ji}}{2}; \quad B_{ij} = \left(1 + \sum_{k \neq i,j} G_i(\theta_{ijk}) f_{ik}(r_{ik})\right)^{-\delta} \\
G(\theta) &= a_o \left\{1 + \frac{c_o^2}{d_o^2} - \frac{c_o^2}{d_o^2 + (1+\cos\theta)^2}\right\}
\end{aligned}
\tag{3.1}
$$

where E is the potential energy, V_R is the repulsive potential, V_A is the attractive potential, B_{ij} is the many-body term, f_{ij} is the cut-off function, i, j and k are atom indices, r is the radial distance between atom i and atom j, θ_{ijk} is the angle between atom i, atom j, and atom k, and D_{ij}, S_{ij}, β_{ij}, R_{ij}, $R_{ij}^{(1)}$, $R_{ij}^{(2)}$, a_o, c_o, and d_c are constants [91].

A major contribution of this thesis is the implementation of the Brenner potential for LAMMPS molecular dynamics software. The code for the Brenner potential is adapted from a similar code by Henry [100] and located in Appendix D.

3.2 Verlet Integrator Algorithm

LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) is an open-source code developed by Sandia National Laboratories for MD computation on parallel processors [107]. LAMMPS allows the ensemble to evolve by solving Newton's equations with a velocity-verlet integrator and updating each atom position, $x(t)$ and velocity, $\dot{x}(t)$ at every timestep, Δt. The velocity-verlet algorithm calculates the updated position and current velocity through Taylor expansions [50].

$$\begin{aligned} x(t+\Delta t) &= x(t) + \Delta t \cdot \dot{x}(t) + \frac{1}{2}\Delta t^2 \cdot \ddot{x}(t) + O(\Delta t^3) \\ x(t-\Delta t) &= x(t) - \Delta t \cdot \dot{x}(t) + \frac{1}{2}\Delta t^2 \cdot \ddot{x}(t) - O(\Delta t^3) \end{aligned} \tag{3.2}$$

Combining these equations yields the updated position and current velocity [50].

$$\begin{aligned} x(t+\Delta t) &= 2x(t) - x(t-\Delta t) + \Delta t^2 \cdot \ddot{x}(t) + O(\Delta t^4) \\ \dot{x}(t) &= \frac{x(t+\Delta t) - x(t-\Delta t)}{2\Delta t} + O(\Delta t^2) \end{aligned} \tag{3.3}$$

3.3 Statistical Ensembles

Macroscopic thermodynamic quantities are ascribed to an MD ensemble through statistical mechanics [50]. The microcanonical ensemble or NVE ensemble is composed of a probability distribution of microstates (e.g. atomic positions and velocities) which average to macro thermodynamics properties (e.g. system temperature and pressure) for a fixed system volume and total energy [50]. It is important to average the macrostate of interest over a sufficiently large space-time sample of microstates in order to measure the

correct thermodynamic properties. In addition to the NVE ensemble, the isothermal-isobaric or NPT ensemble is used as a means of equilibrating the system to a set pressure and temperature, while allowing the macrostates (e.g. energy and volume) to vary. The NPT ensemble is implemented through the Nosé-Hoover thermostat [108] and barostat [109].

3.4 Boundary Conditions

The MD system evolves through integration of Newton's second law for a given set of forces described by interatomic potentials and point masses, and a fixed statistical ensemble. The solution of these differential equations requires both initial conditions and boundary conditions. The initial conditions are atomic positions and velocities. The available boundary conditions in LAMMPS are either periodic or fixed [107]. If an atom passes through a periodic boundary, it exits and re-enters through the opposing boundary. If an atom passes through a fixed boundary, it is lost from the simulation. Periodic boundary conditions are analogous to Born-von Karman boundary conditions in lattice dynamics where for a linear chain of N atoms (Figure 2-5), atom number N+1 is equal to atom 1 (i.e. the end of the chain is the beginning of the chain) [46].

3.5 Non-equilibrium Molecular Dynamics

Non-equilibrium molecular dynamics (NEMD) is a simulation in the microcanonical ensemble where a gradient or flux is imposed to observe transport properties. For simulations of thermal properties, either a temperature gradient or heat flux is imposed on the equilibrium system to observe the ensemble thermal conductivity

or thermal interface resistance [57, 58, 77, 84, 88, 96, 110-115]. Jund and Jullien developed a simple method for imposing a heat flux and observing the resultant temperature drop to measure the thermal conductivity of vitreous silica [61]. In this method, the velocities, v in a heated section is scaled up and the velocities in a cooled section are corresponding scaled down for a fixed amount of heat, Q as seen in Figure 3-1 and Eq (3.4) [61].

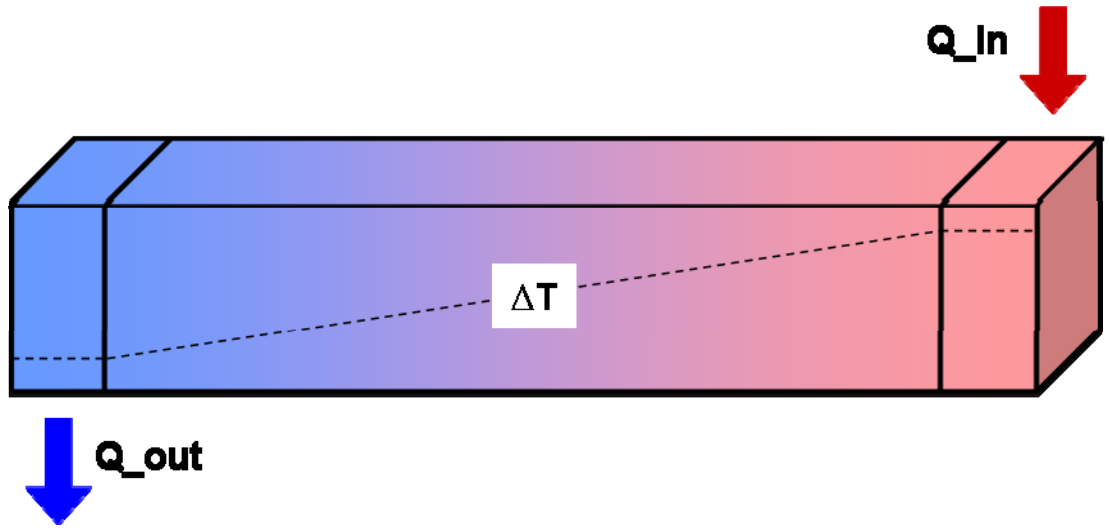

Figure 3-1. Jund-Jullien Method

$$v_i = v_{cm} + \alpha(v_i - v_{cm})$$
$$\alpha = \sqrt{1 \pm \frac{Q}{E}} \qquad (3.4)$$
$$E = \frac{1}{2}\sum_i m_i {v_i}^2 - \frac{1}{2}\sum_i m_i {v_{cm}}^2$$

where v_i is the atomic velocity in the heated or cooled region, v_{cm} is the velocity of the center of mass of the heated or cooled region, m_i is the atomic mass, α is the scaling factor, Q is the heat added or subtracted, and E is the thermal kinetic energy [61].

3.6 Phonon Properties

Monatomic Lattice

In addition to equilibrium thermodynamic properties and non-equilibrium transport properties, MD simulation may also be used to calculate the phonon density of states and dispersion curves for an ensemble lattice [66, 82, 85, 116, 117]. The density of states is calculated from the Fourier transform of the velocity autocorrelation function as shown in Eq. (3.5) [85].

$$D(\omega) = FFT(<\vec{v}(t)\cdot\vec{v}(0)>) \tag{3.5}$$

This definition of phonon density of states is valid for the harmonic approximation [117]. Since the MD system is fully anharmonic, the harmonic DOS of an anharmonic system is referred to as quasi-harmonic. Dickey and Paskin [116] provide a derivation of Eq. (3.5), and an implementation of Eq. (3.5) for a LAMMPS output file is located in the Appendix.

The phonon dispersion curves are calculated by taking the 2D Fourier transform of the first Brillouin zones in space and in time as shown in Eq. (3.6)

$$E(\omega,k) = \frac{1}{3n}\sum_{p}^{3}\sum^{n}\left|\frac{1}{N}\int v_p(z,t)e^{i(kz-\omega t)}dtdz\right|^2 \tag{3.6}$$

where $E(\omega,k)$ is the energy density, n is the number of atoms per unit cell, p is the polarization, N is the number of unit cells in the z-direction, v_p is the velocity of a polarization p, k is the wavevector, ω is the frequency, t is the time coordinate, and z is the spatial coordinate [82].

Figure 3-2 is the phonon density of states and dispersion curve for a one-dimensional, monatomic lattice of carbon atoms with Brenner interatomic potential

calculated with molecular dynamics and Eq. (3.5) and (3.6). This computational result is comparable to the analytical solution from Section 2.2.

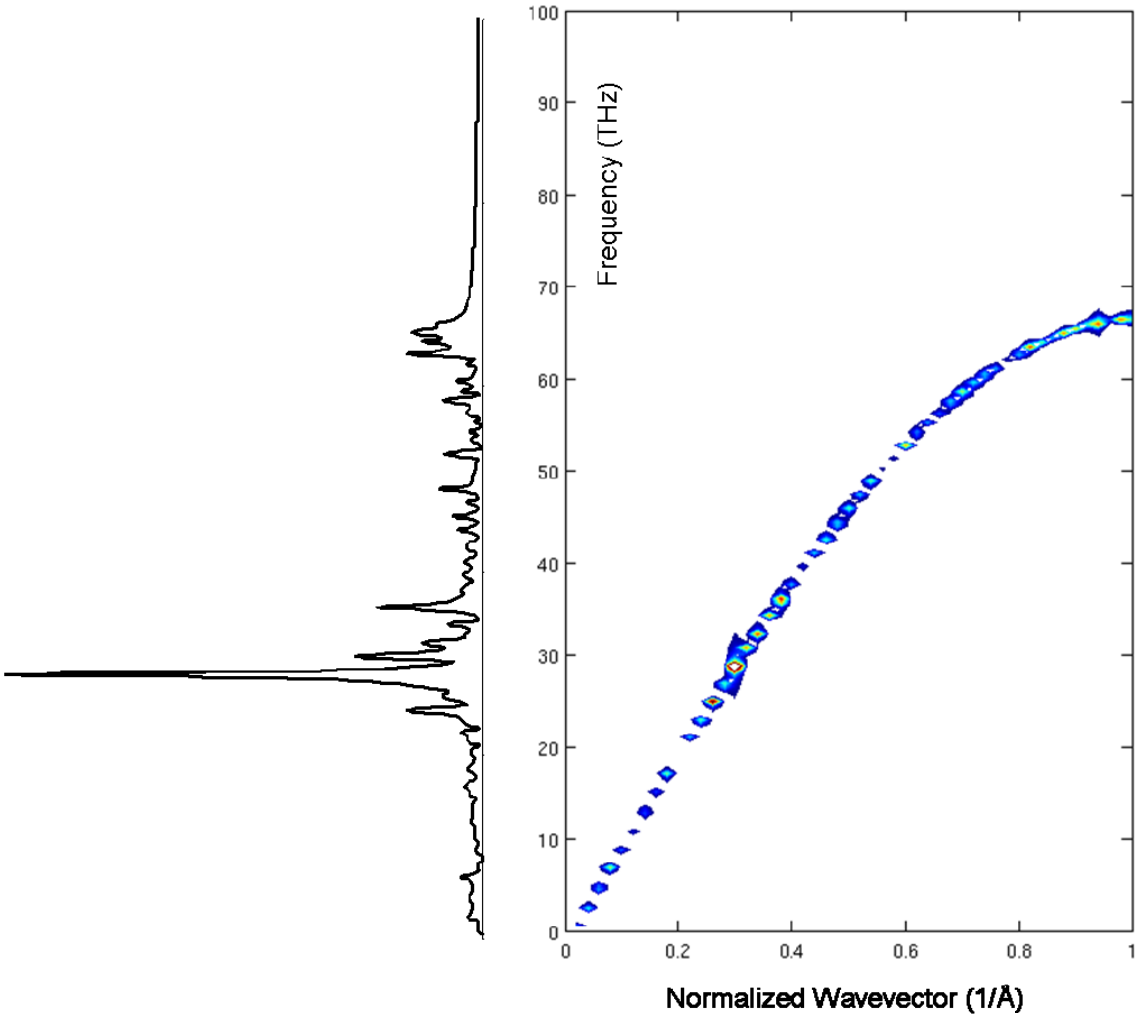

Figure 3-2. 1D Chain DOS and Dispersion from MD at 10K

The MD results in Figure 3-2 are the phonon properties of the classical-anharmonic system as opposed to the classical-harmonic system in Figure 2-6 and Figure 2-10. The anharmonicity of the MD lattice results in a concentration of phonon modes around 30 THz. The anharmonic MD dispersion relation is shaped similarly to the

harmonic dispersion relation in Figure 2-6 and the discretization of the wavevector due to lattice spacing is clearly illustrated in Figure 3-2.

CHAPTER 4. PHONON TRANSMISSION MODELS

The flow of heat across a material interface causes a discontinuous change in temperature at the interface [118, 119]. This resistance is known as the Kapitza resistance or inversely, the Kapitza conductance as in Eq. (4.1).

$$q'' = \sigma_K \cdot \Delta T \tag{4.1}$$

where q is the heat flux per unit area, σ is the Kapitza conductance, and ΔT is the temperature change at the interface. Heat is conducted across a material interface through the quantum transmission of phonons and electrons through phonon-phonon, electron-electron, or electron-phonon processes.

4.1 Phonon Mismatch Models

The flow of heat across an interface from material 1 to material 2 may be expressed in terms of phonon transmission as

$$\dot{Q}_{1\to 2}(T_1) = \frac{1}{2}\sum_j \int_A \int_{\theta=0}^{\frac{\pi}{2}} \int_{\omega=0}^{\omega_d} D_{1,j}(\omega)\cdot n(\omega,T) \cdot \hbar\omega \cdot c_{1,j} \cdot \alpha_{1\to 2}(\omega)\cdot \cos(\theta)\sin(\theta) d\omega d\theta dA \tag{4.2}$$

where Q is the heat flux, D is the density of states, n is the occupation number, h is Planck's constant, ω is the frequency, A is the interface area, θ is the phonon-interface angle, c is the phonon mode speed, α is the transmission coefficient, and j is the phonon mode [119]. This general form of phonon transmission is known as the acoustic mismatch model (AMM) [119-121]. The AMM assumes phonons are reflected or transmitted according to the difference in acoustic impedances of the two materials. The

assumption of perfect specular reflection or transmission means there is no scattering at the interface as shown in Figure 4-1.

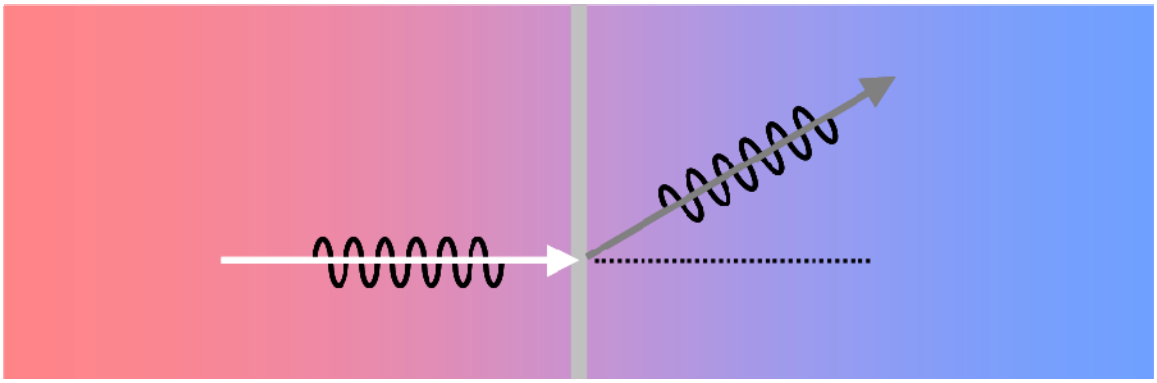

Figure 4-1. AMM – Specular Transmission

Another way to solve Eq. (4.2) is to assume complete diffuse scattering at the interface. The diffuse mismatch model (DMM) assumes that all of the phonons scatter diffusely and "forget" their origin at the interface [119-121]. The diffuse phonons then emerge with no directional preference into either material with the probability given by the relative densities of states of the two materials. Under this assumption, the transmission coefficients may be expressed as

$$\alpha_{1\to 2}(\omega)+\alpha_{2\to 1}(\omega)=1$$
$$\alpha_{1\to 2}(\omega)=\frac{\sum_j D_{2,j}(\omega)c_{2,j}}{\sum_j D_{1,j}(\omega)c_{1,j}+\sum_j D_{2,j}(\omega)c_{2,j}} \tag{4.3}$$

and the transmission equation may be written without angle dependence since the scattering is diffuse and not specular [119]. An illustration of diffuse scattering is shown in Figure 4-2.

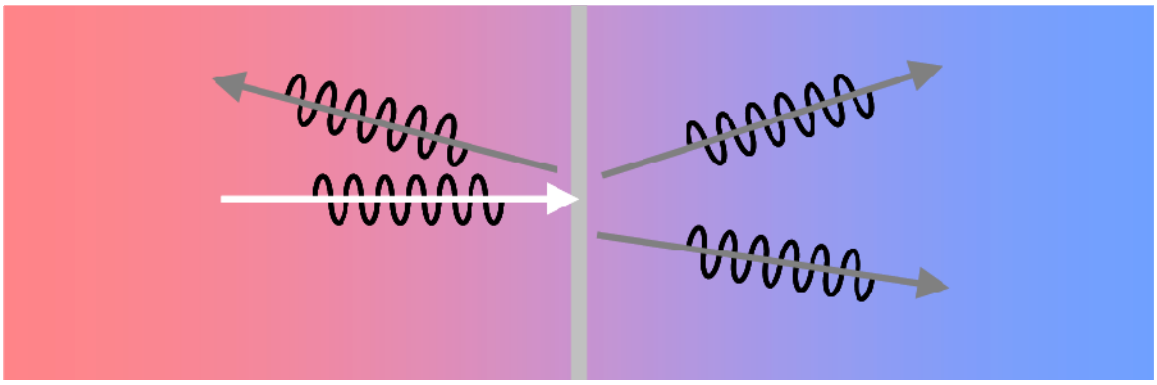

Figure 4-2. DMM – Diffuse Scattering

The Kapitza conductance is expressed as [119, 121]

$$\sigma_K(T) = \frac{\dot{Q}_{2\to1}(T_2) - \dot{Q}_{1\to2}(T_1)}{A(T_2 - T_1)} \tag{4.4}$$

Rewriting Eq. (4.2) and Eq. (4.4) with the DMM assumptions (Eq. (4.3)), the Kapitza conductance is [122]

$$\sigma_K(T) = \frac{1}{4}\sum_j \int_0^{\omega_{\max}} D_{1,j}(\omega)\hbar\omega c_{1,j}\frac{\partial n(\omega,T)}{\partial T}\alpha_{1\to2}(\omega)d\omega \tag{4.5}$$

The AMM and DMM theories are limiting cases of the nature of interface phonon transmission: completely specular refraction without scattering or completely diffuse scattering without angular dependence. Both theories assume elastic scattering of phonons meaning that each phonon mode only interacts with the corresponding mode in the other material. A given phonon frequency may only emit a phonon of the same frequency at the interface [113, 122-125]. For highly mismatched materials, there may be a gap between the phonon spectra of the two materials resulting in a limited number of phonons available for elastic scattering as a percentage of the total phonon population. In

these circumstances phonons may scattering inelastically as well as elastically, providing an additional channel for thermal transport [113, 122-125].

4.2 Lattice Dynamics

Lattice dynamical calculations have been used to study phonon transmission and scattering for classical harmonic lattices with simple elastic interfaces [126-130]. Stoner and Maris compared lattice dynamical calculations to experimental measure for the Kapitza conductance of metals (lead, gold, aluminum and titanium) on dielectrics (barium fluoride, diamond, and sapphire) [128]. When compared to the experimental measurements, lattice dynamical results for the lead-diamond interface suggest that inelastic phonon scattering and lattice anharmonicity may significantly contribute to the Kapitza conductance [128]. Young and Maris studied two face-centered cubic lattices with various masses, stiffness, and temperatures [129]. Kapitza conductance is shown to be dependent on the lattice mass, stiffness, and temperature but not on the stiffness of the interface bonds if they are weaker than the stiffest bond and stronger than the weakest bond [129].

Panzer and Goodson studied the lattice dynamics of a one-dimensional chain in contact with a two-dimensional or three-dimensional semi-infinite lattice [126]. Results show incomplete transmission even for lattices with the same material properties, indicating the presence of a strong geometric effect on Kapitza conductance [126]. Transmission from the one-dimensional chain to the two-dimensional lattice was greater than the transmission from the one-dimensional chain to the three-dimensional lattice for all longitudinal mode frequencies [126]. The transverse modes of the one-dimensional

were mostly reflected and not transmitted into the lattice and the optimal configuration was not a matching material parameter set but rather a large impedance for the one-dimensional chain and a lower impedance for the higher dimension lattice [126].

4.3 Molecular Dynamics Simulation

Unlike lattice dynamical calculations, MD simulation allows for fully anharmonic lattices and inelastic phonon interface scattering. Schelling *et al.* have developed a wave packet method to study phonon transmission through interfaces [131]. A Gaussian wave packet is set as part of the initial conditions in MD simulation at zero degrees K. As the ensemble evolves, the wave propagates towards the interface. After the wave collides with the interface, the energy of transmission and reflection is measured to calculate the transmission coefficient. When coupled with NEMD simulation the wave packet method is a useful qualitative tool to estimate the modal phonon transmission [111, 114, 131-133].

Stevens *et al.* used the diffuse mismatch model with scaled phonon densities of states from MD simulation [113, 125]. The Kapitza conductance was shown to significantly depend on temperature, indicating a large inelastic phonon scattering transmission. Modifications to the DMM theory to incorporate inelastic phonon scattering as a function of coupling to local or joint densities of state at the interface have been proposed [122, 124, 134-136].

4.4 Carbon Nanotube – Substrate

Prasher has developed a combined thermal theory of the carbon nanotube thermal conductivity, interface resistance, and constriction resistance [30, 31, 137, 138]. Cola *et al.* used combined the theoretical models of Prasher with a contact mechanics model for estimating the area of contact for an individual nanotube on a substrate in a CNT TIM [139]. Using these combined models, Cola estimates a CNT-copper thermal interface resistance of 0.003 mm^2w/K [139].

CHAPTER 5. RESULTS

This chapter contains the results of non-equilibrium molecular dynamics simulation and diffuse mismatch model calculations for nanotube-substrate interfaces at various temperatures and with varying degrees of interface disorder. Example LAMMPS scripts and temperature profiles are located in the Appendix. The results include a system temperature profile, NEMD Kapitza or interfacial conductance calculation using the Jund-Jullien method [61], densities of states, transmission coefficients, and diffuse mismatch model results.

5.1 Ideal Interfaces – Large System

In order to estimate the degree of inelastic scattering and lattice anharmonicity, the thermal interface conductance is calculated as a function of temperature. First, the system is equilibrated to the set temperature using a Nose-Hoover thermostat. Second, the system is allowed to evolve in the microcanonical ensemble where the phonon dynamics are recorded. Third, heat is added to the center of the nanotube and removed from the end of the substrate block according to the Jund-Jullien method as shown in Figure 5-1. The system is allowed to reach equilibrium and an averaged temperature profile is calculated over the final 100 ps. Temperatures are calculated from the atomic velocities according to the equipartition theorem (Eq. (2.26)). Finally the phonon densities of states are used to calculate the thermal interface conductance according to the diffuse mismatch model (Eq. (4.5)). The parameters for each simulation are listed in Table 5-1. Temperature profiles are displayed in Appendix A.

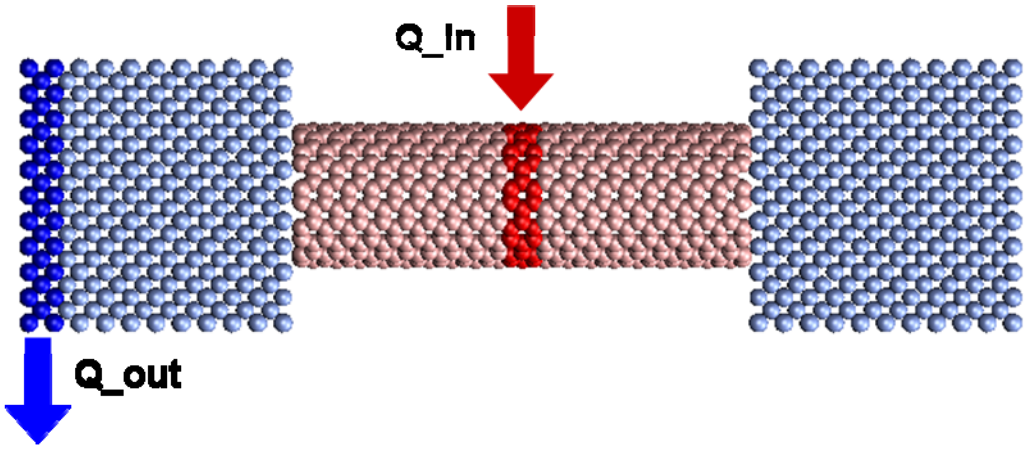

Figure 5-1. NEMD Nanotube-Substrate Configuration

	Silicon	Diamond	Copper
Substrate Mass	28.0855 amu	12.0107 amu	63.55 amu
Block Size	9x19x9	9x19x9	9x14x9
Nanotube Length	200A	200A	100A
Chirality	(10,10)	(10,10)	(10,10)
Nanotube Mass	12.0107 amu	12.0107 amu	12.0107 amu
Block 1 Count	5544	12474	4698
Block 2 Count	5472	12312	4536
Nanotube Count	3220	3220	1620
Total Count	14236	28006	10854
Boundary Condition	Periodic	Periodic	Periodic
Interatomic Potentials	Si-Si Brenner	C-C Brenner	C-C Brenner
	C-C Brenner		Cu-Cu MEAM
	Si-C Brenner		Cu-C LJ
Time Step	0.0005 ps	0.0005 ps	0.0005 ps
NPT Time	25 ps	25 ps	25 ps
NVE Time	100 ps	100 ps	100 ps
HEAT Time	150 ps	150 ps	150 ps
Spatial Average Step	5.0 A	3.0 A	3.0 A
Time Average Step	5 ps	5 ps	5 ps

Table 5-1. Parameters for Large System MD

In addition to energy-based NEMD calculations, wave-based DMM calculations are also performed for each simulated material and temperature. The calculations are described in Chapter 4 and Figure 5-2, Figure 5-3, and Figure 5-4 are included to illustrate the calculations for each material type across each frequency spectra.

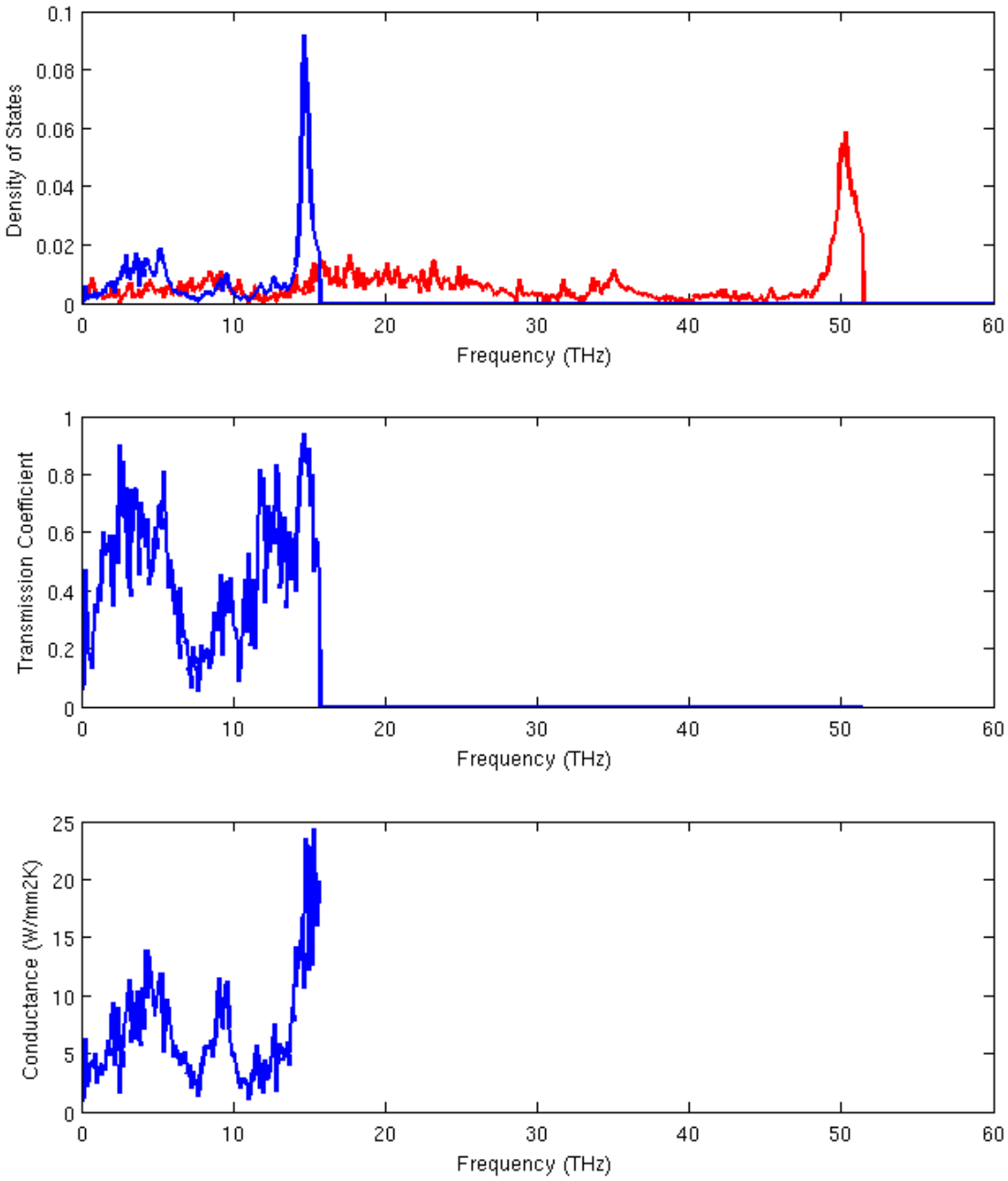

Figure 5-2. DOS and DMM for Silicon (blue)-CNT (red) at 300K

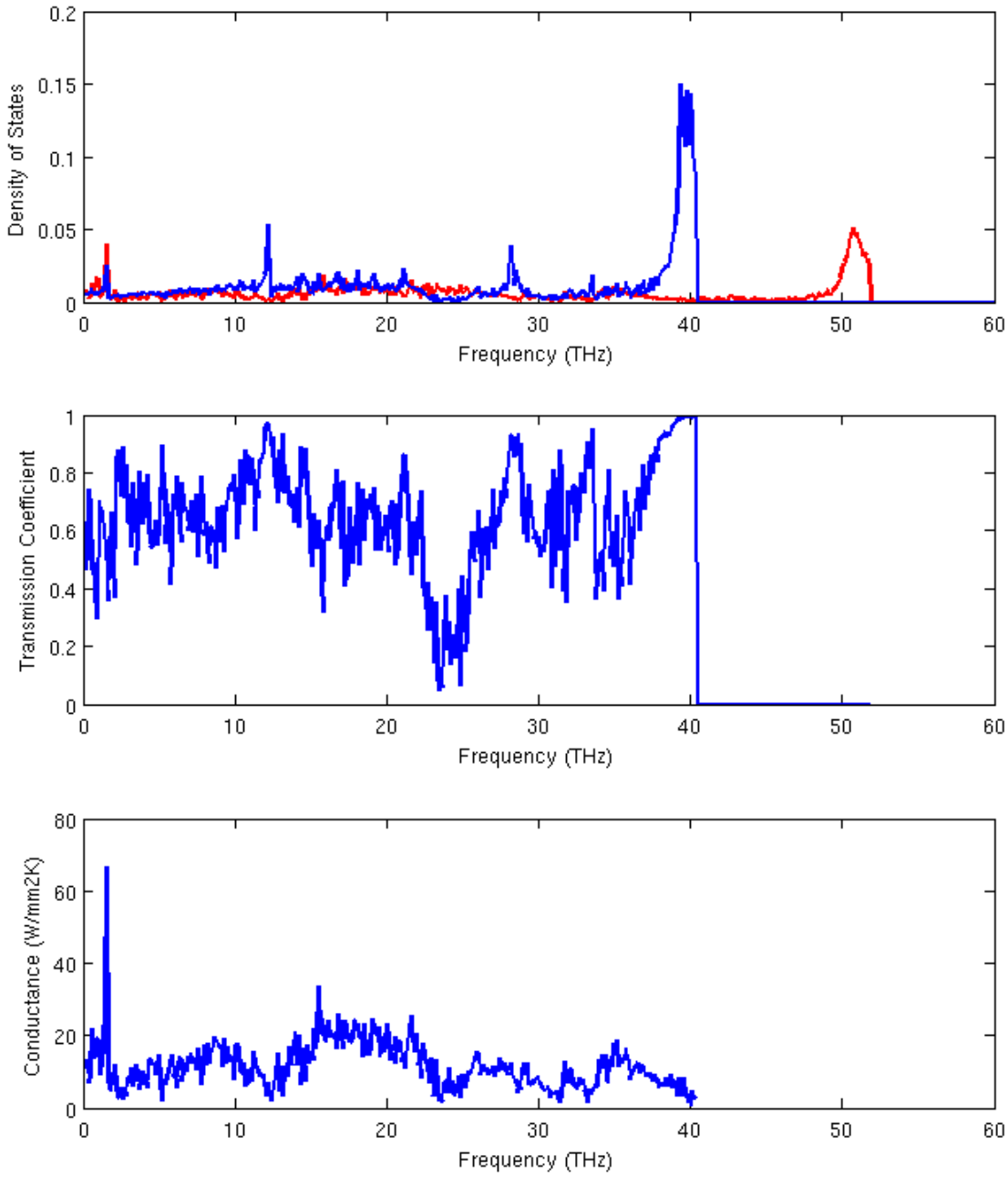

Figure 5-3. DOS and DMM for Diamond Carbon (blue)-CNT (red) at 300K

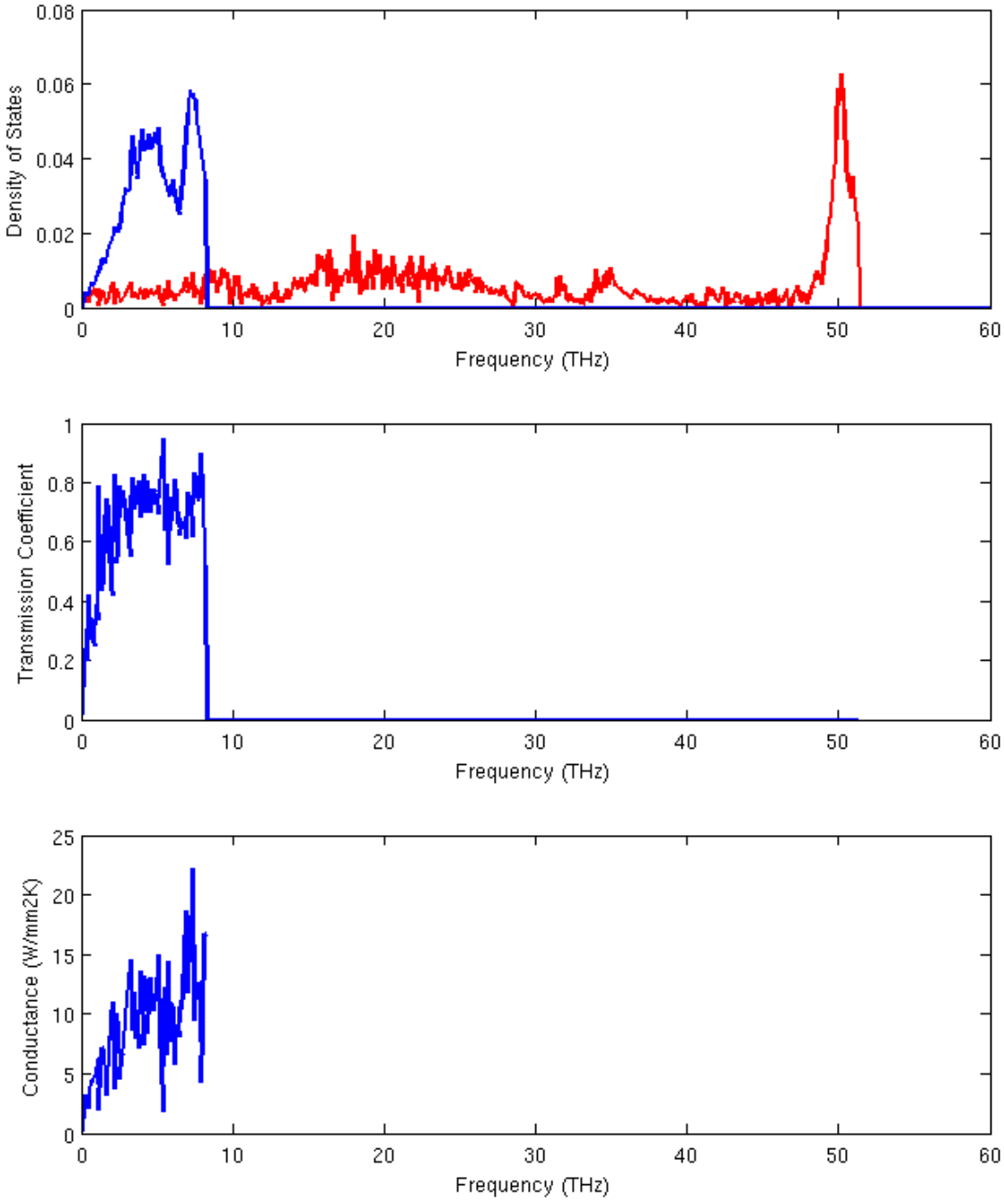

Figure 5-4. DOS and DMM for Copper (blue)-CNT (red) at 300K

The complete set of results of both NEMD and DMM calculations are summarized in Table 5-2 where Q is the imposed heat flux, ΔT is the temperature drop, σ is the thermal interface conductance from NEMD, and DMM is the thermal interface conductance from the wave-based diffuse mismatch calculations. These results are displayed graphically in Figure 5-5. Note that both the magnitude of thermal interface conductance and the trend with temperature are equally reflected by both models. This correspondence between models demonstrates that the phonon transport mechanism resulting in thermal interface conductance may be accurately described as diffuse elastic scattering for all three materials.

Silicon				
Temperature (K)	Q (W/mm^2)	ΔT (K)	σ (W/mm^2K)	DMM (W/mm^2K)
100	1.71E+04	28.6	6.00E+02	8.21E+02
300	3.43E+04	40.1	8.55E+02	1.10E+03
500	5.72E+04	45.3	1.26E+03	1.21E+03
700	5.72E+04	45.7	1.25E+03	1.24E+03
900	5.72E+04	46.7	1.22E+03	1.29E+03
Diamond				
Temperature (K)	Q (W/mm^2)	ΔT (K)	σ (W/mm^2K)	DMM (W/mm^2K)
100	1.71E+04	9.6	1.79E+03	2.94E+03
300	6.86E+04	19.1	3.59E+03	4.72E+03
500	1.14E+05	28.2	4.05E+03	4.88E+03
700	1.14E+05	28.2	4.05E+03	4.92E+03
900	1.14E+05	26.9	4.25E+03	5.16E+03
Copper				
Temperature (K)	Q (W/mm^2)	ΔT (K)	σ (W/mm^2K)	DMM (W/mm^2K)
100	1.14E+04	43.9	2.60E+02	6.29E+02
300	1.14E+04	29.2	3.91E+02	7.65E+02
500	1.43E+04	27.1	5.27E+02	8.96E+02
700	1.71E+04	37.3	4.60E+02	8.82E+02
900	1.71E+04	31.9	5.37E+02	9.78E+02

Table 5-2. Summary of NEMD and DMM for Table 5-1

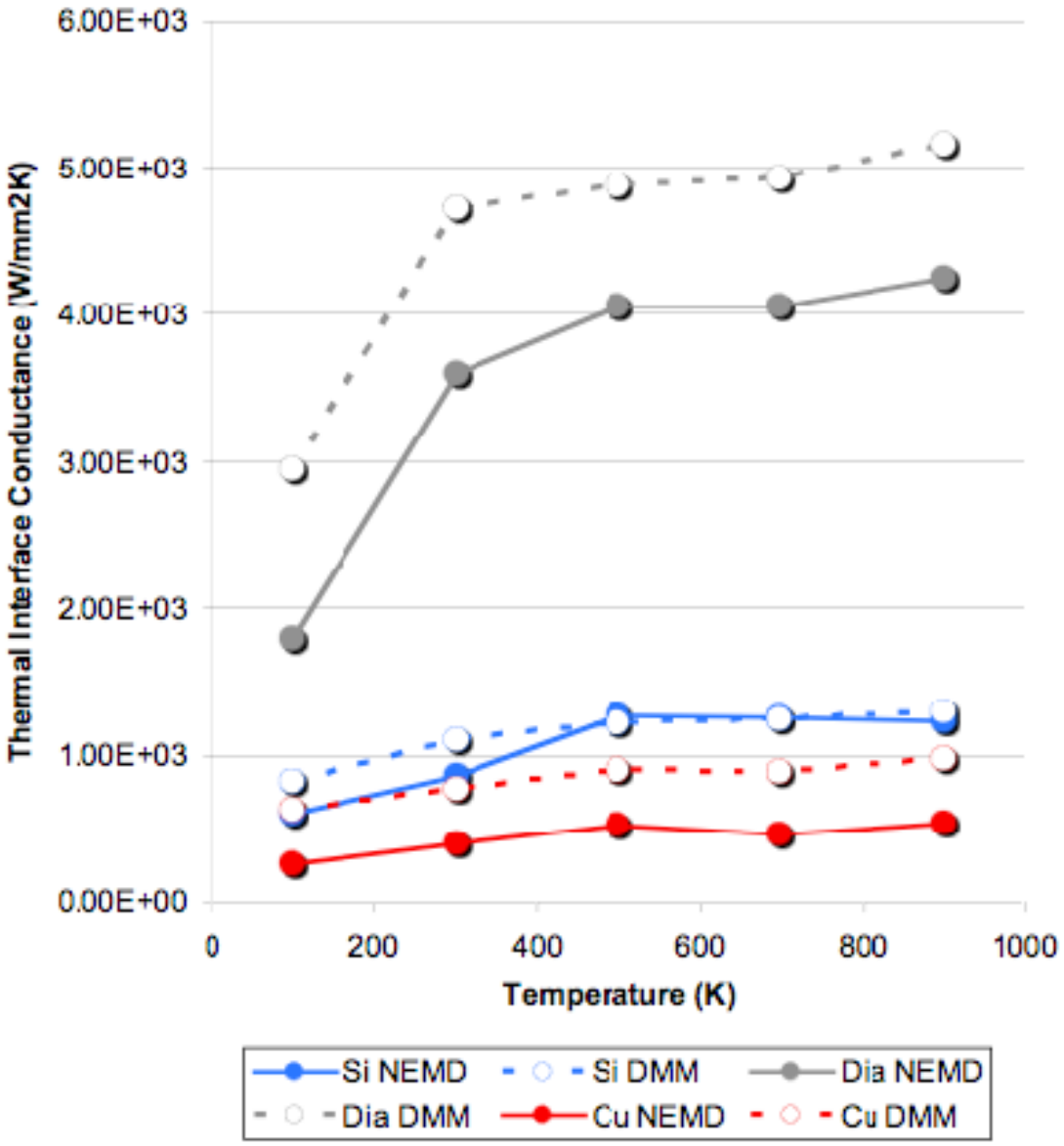

Figure 5-5. NEMD and DMM Results for Table 5-2

5.2 Ideal Interfaces – Small System

Section 5.2 uses the same procedure of to calculate thermal interface conductance as Section 5.1 however with different simulation parameters and a smaller system size. The small system parameters are given in Table 5-3.

	Silicon	Diamond	Copper
Substrate Mass	28.0855 amu	12.0107 amu	63.55 amu
Block Size	5x5x5	7x7x7	7x7x7
Nanotube Length	50A	50A	50A
Chirality	(10,10)	(10,10)	(10,10)
Nanotube Mass	12.0107 amu	12.0107 amu	12.0107 amu
Block 1 Count	1050	2842	1470
Block 2 Count	1000	2744	1372
Nanotube Count	780	780	780
Total Count	2830	6366	3622
Boundary Condition	Periodic	Periodic	Periodic
Interatomic Potentials	Si-Si Brenner	C-C Brenner	C-C Brenner
	C-C Brenner		Cu-Cu LJ
	Si-C Brenner		Cu-C LJ
Time Step	0.0005 ps	0.0005 ps	0.0005 ps
NPT Time	25 ps	25 ps	25 ps
NVE Time	25 ps	25 ps	25 ps
HEAT Time	250 ps	250 ps	250 ps
Spatial Average Step	2.0 A	2.0 A	2.0 A
Time Average Step	25 ps	25 ps	25 ps

Table 5-3. Small System NEMD Parameters

Silicon				
Temperature (K)	Q (W/mm^2)	ΔT (K)	σ (W/mm^{2}K)	DMM (W/mm^{2}K)
100	2.86E+04	138.6	2.06E+02	8.52E+02
300	5.72E+04	108.6	5.26E+02	1.05E+03
500	5.72E+04	65.88	8.68E+02	1.14E+03
700	5.72E+04	39.4	1.45E+03	1.25E+03
900	5.72E+04	25.1	2.28E+03	1.38E+03
Diamond				
Temperature (K)	Q (W/mm^2)	ΔT (K)	σ (W/mm^{2}K)	DMM (W/mm^{2}K)
100	2.86E+04	50.9	5.61E+02	2.75E+03
300	5.72E+04	38.5	1.48E+03	4.65E+03
500	5.72E+04	15.8	3.62E+03	4.93E+03
700	5.72E+04	13.7	4.17E+03	5.14E+03
900	5.72E+04	2.8	2.04E+04	5.13E+03
Copper				
Temperature (K)	Q (W/mm^2)	ΔT (K)	σ (W/mm^{2}K)	DMM (W/mm^{2}K)
100	3.43E+04	183.4	1.87E+02	6.70E+02
300	3.43E+04	136.9	2.50E+02	9.89E+02
500	3.43E+04	124.7	2.75E+02	8.43E+02
700	3.43E+04	102.5	3.35E+02	1.11E+03
900	3.43E+04	120.5	2.85E+02	1.09E+03

Table 5-4. Summary of NEMD and DMM for Table 5-3

The results in Table 5-4 are compared graphically to the results in Table 5-2 in Figure 5-6, Figure 5-7, and Figure 5-8 to illustrate the effect of simulated system size or number of atoms on both NEMD and DMM calculations. Unlike the large systems in Figure 5-5, the DMM model does not predict the thermal interface conductance temperature trend of smaller simulated systems. For small silicon and diamond carbon systems, the NEMD temperature dependence strengthens with increasing temperature as opposed to the DMM temperature dependence which levels off. This discrepancy suggests that for small systems sizes, there an inelastic transport channel not found in larger simulation system sizes. Therefore unlike the larger NEMD results which are fully

predicted by the diffuse elastic scattering mechanism of the DMM, the smaller NEMD results are both under predicted and overpredicted by the DMM due to the presence of inelastic interface scattering either enhancing or inhibiting thermal transport.

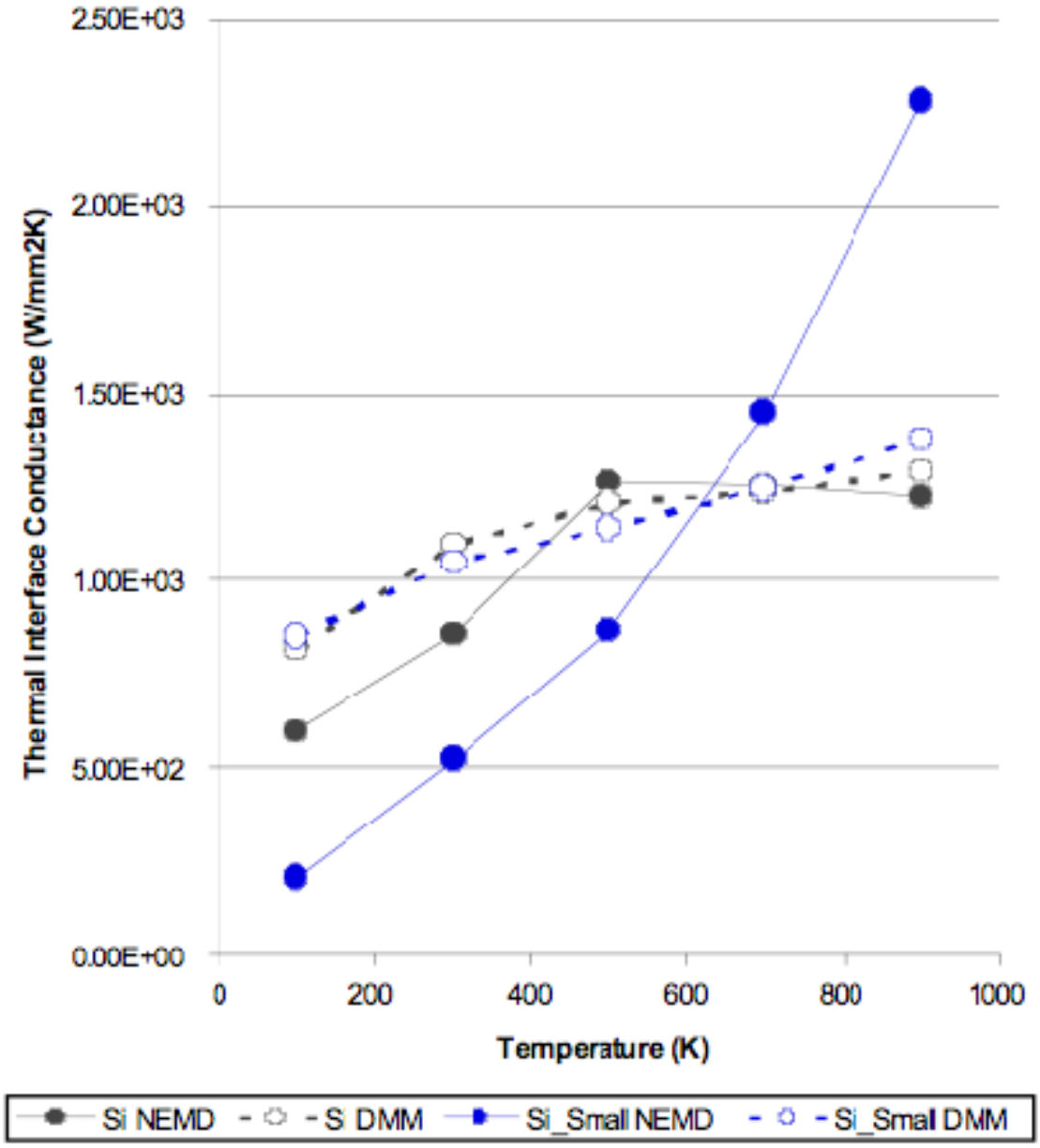

Figure 5-6. Small vs. Large Systems for Silicon NEMD and DMM

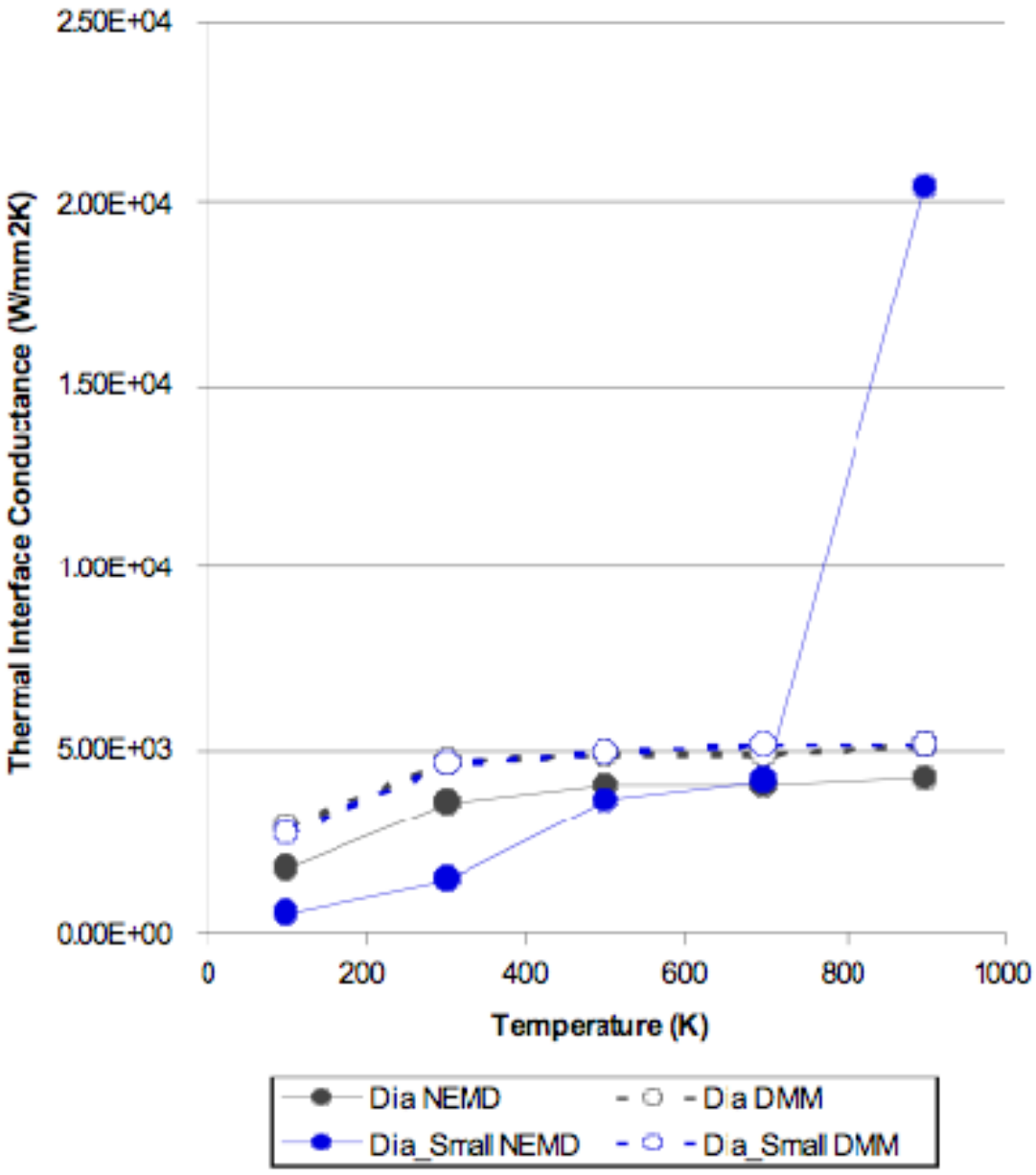

Figure 5-7. Small vs. Large Systems for Carbon NEMD and DMM

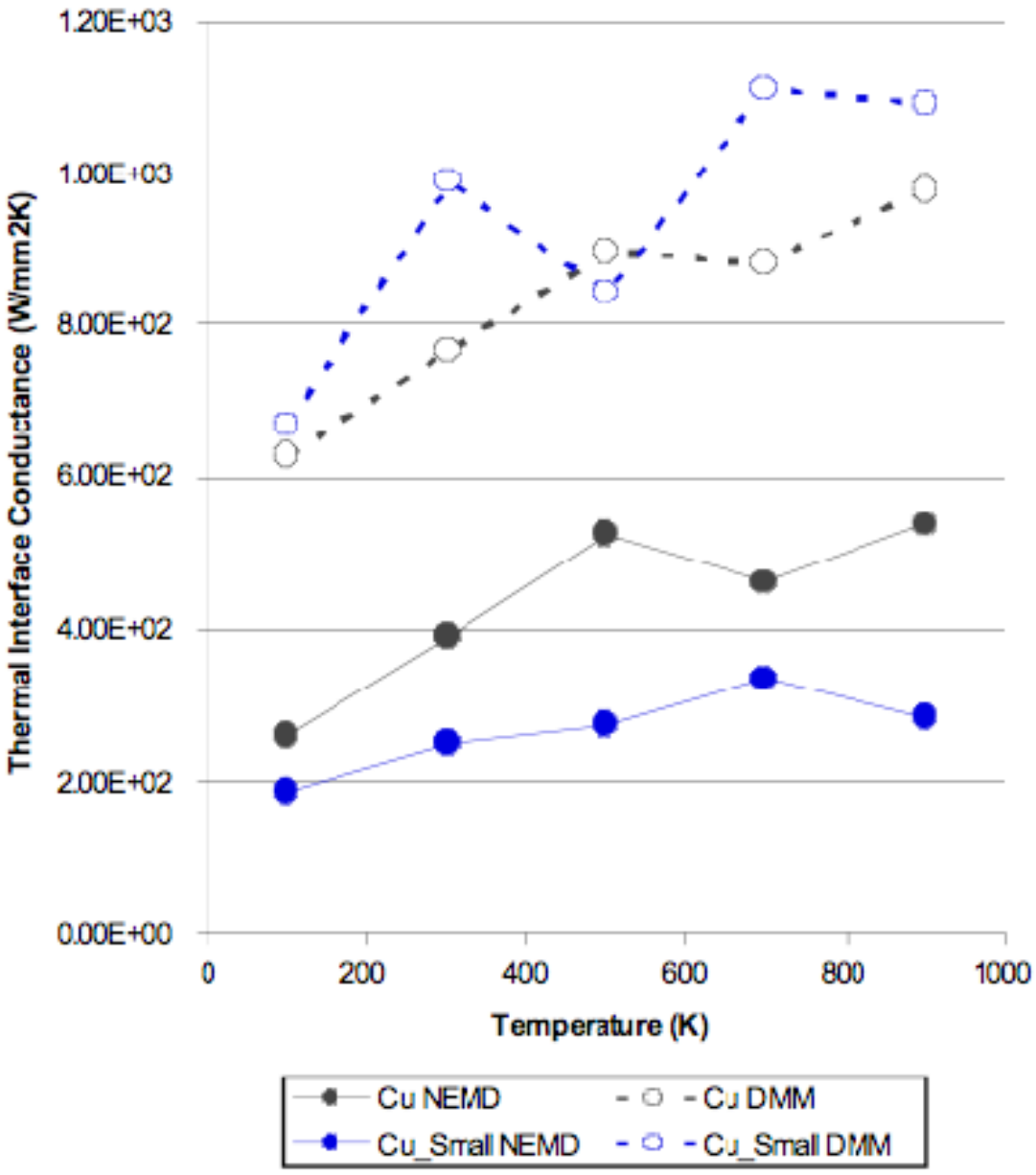

Figure 5-8. Small vs. Large Systems for Carbon NEMD and DMM

5.3 Non-Ideal Interfaces

Section 2.1 and 2.3 describe ideal interfaces where the carbon nanotube lattice and substrate lattices are in perfect contact. Since many interfaces may contain some degree of disorder, various non-ideal interfaces are considered using the same simulation parameters from the silicon-carbon nanotube 300K case in Table 5-1. Cases considered are capped vs. uncapped nanotubes, side contact instead of on end contact as shown in Figure 5-10, 2x1 silicon surface reconstruction, multiwalled CNT (MWCNT) with and without intertube interaction through van der Waals forces, physical bonding instead of chemical bonding through van der Waals forces, a 1000 Å nanotube length instead of 200 Å, and a thermal rectification example where the heat flows into the nanotube from the silicon. The results of these cases are shown in Figure 5-9.

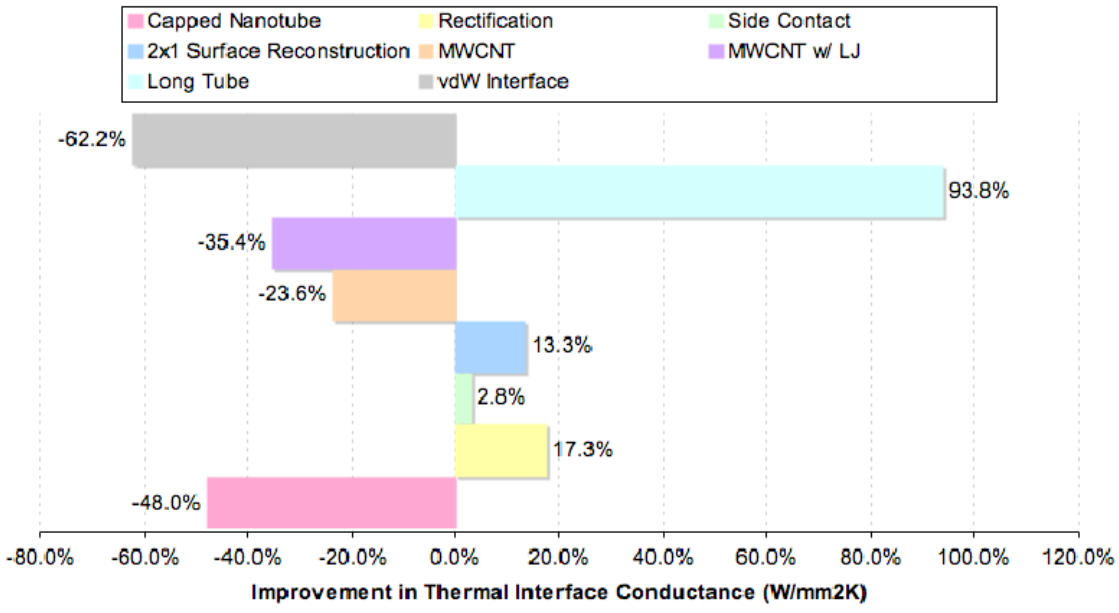

Figure 5-9. NEMD for Non-Ideal Interfaces

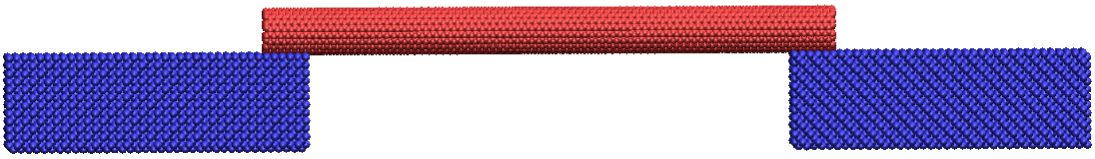

Figure 5-10. End Contact Configuration

The reduction in thermal interface conductance for the physical van der Waals interface may results from the damping of transport channels due to the weakened interface bonding. For the long tube case, increase in thermal interface conductance may be due to the increase in low frequency, long wavelength modes in the nanotube coupling to modes in the silicon blocks. For both of these cases, the area of contact used to define heat flux is the same.

For the multiwalled case, the area is defined as the superposition of the (5,5), (10,10), and (15,15) carbon nanotube rings of van der Waals thickness. Both interacting and non-interacting MWCNTs are considered. In both cases, the thermal interface conductance is reduced. This may be a function of the differing acoustic properties of each nanotube size or a size artifact of the fixed simulation cell size with no physical meaning. There may also be a constriction or spreading resistance between the nanotube and substrate. Interaction between nanotubes further reduces thermal interface conductance either through phonon scattering or modification of the densities of states of each tube.

For the capped case, the area of contact is defined as the same as the open ended case for the calculations resulting in an effective reduction in thermal interface conductance due to the smaller contact area of the cap. For the thermal rectification case, the direction of heat flow is reversed so that the heat flows from the silicon into the

carbon nanotube. The result is an increase in thermal interface conductance reaffirming the directional preference of phonon transport described by the diffuse mismatch model.

Two final cases modify only the interface between silicon and carbon nanotube and not the bulk properties or dimensions. In the first case, the ideal silicon surface is replaced by a realistic 2x1 surface reconstruction. In the second case, the end contact configuration is replaced by a side contact configuration (Figure 5-10) where the area of contact is defined as the rectangular projection of the nanotube cross-section onto the silicon block equal to the ring of van der Waals thickness area in the ideal case. For both modified interfaces, the thermal interface conductance improves from the ideal case. This would be counter intuitive if the phonon transport was spectral as in Figure 4-1, but is expected for diffuse phonon transport as in Figure 4-2, where increased disorder at the interface allows for increased scattering of phonons and where transport has no direction dependence.

A more direct way to simulate the effect of interface disorder is to modify the masses of the carbon atoms at the nanotube tips, creating a stiffer or softer interfacial region. The marginal effect of the interfacial region may then be used to draw conclusions about the nature of phonon transport through the silicon – carbon nanotube interface. The modified configuration is shown in Figure 5-11 and the results of mass modification are shown in Figure 5-12.

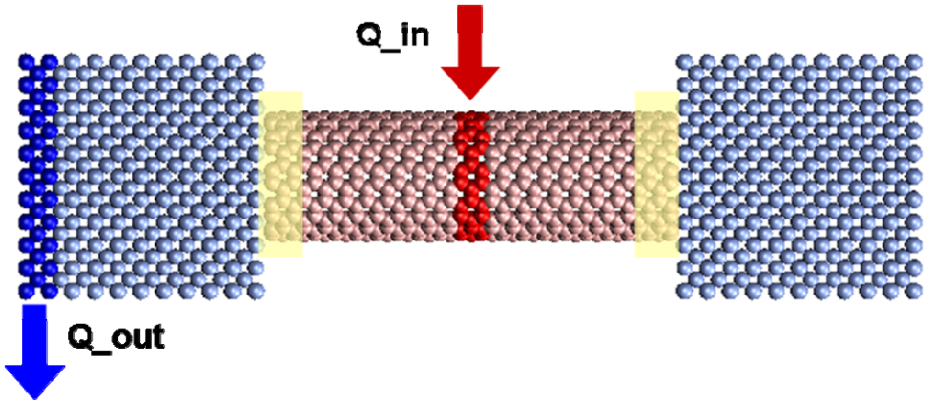

Figure 5-11. Mass Modified Configuration

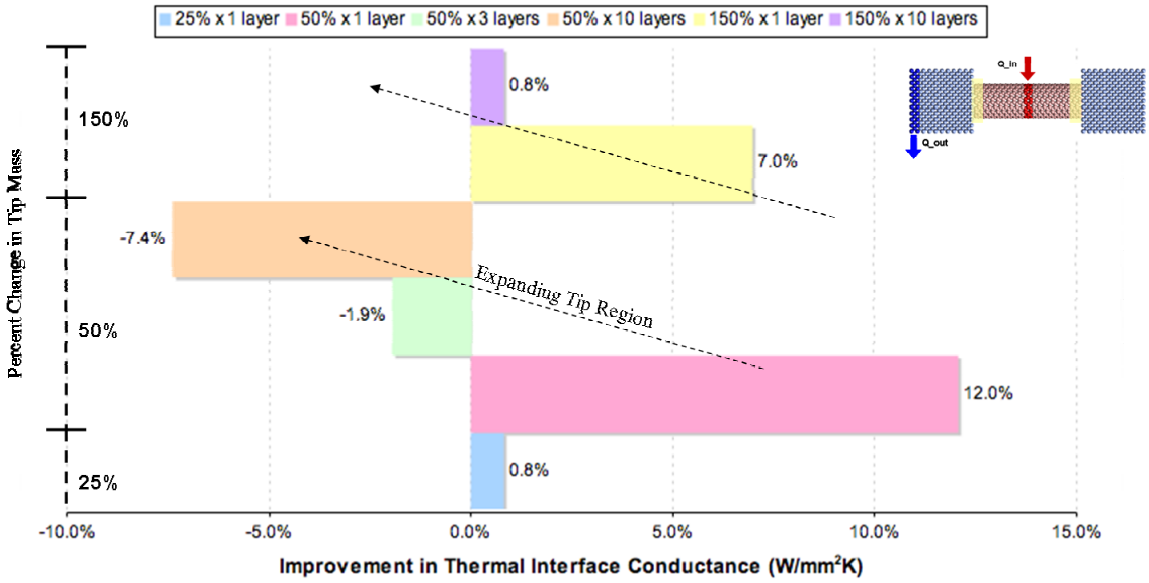

Figure 5-12. Mass Modified NEMD Results

The above results show that for small interfacial regions of both larger and smaller masses, the interface thermal conductance is enhanced. As the size of the interfacial region increases, the interfacial region delocalizes into a third material with two interfaces which inhibits the thermal transport.

CHAPTER 6. CONCLUSIONS AND RECOMMENDATIONS

Conclusions

The carbon nanotube – substrate interface is the most important limiting factor on the performance of carbon nanotube-based thermal interface materials. This thesis has review and implemented the important concepts necessary to computationally evaluate this resistance for various substrate materials, temperatures, and interface conditions. The major contributions of this work are:

- A frequency based diffuse mismatch model with phonon density of states from molecular dynamics simulation accurately predicts the thermal interface resistance from non-equilibrium molecular dynamics simulation. This suggests that the nanotube-substrate interface with both acoustic and geometric mismatch is best described by diffuse-elastic phonon scattering.
- For small simulation sizes, inelastic scattering is evidenced by the difference in temperature dependence between the diffuse mismatch model and the molecular dynamics result.
- Local interface disorder assists the thermal transport. As the region of disorder increases, the thermal transport is inhibited. This suggests that the diffuse scattering is enhanced by disorder that is local to the interface, but as the interface becomes of a third intermediary material, the transport is inhibited.

- The thermal interface resistance of an ideal Si-CNT contact is approximately 0.001 mm^2K/W which is an order of magnitude below current estimates for other single CNT-substrate contacts.

Recommendations

This thesis provides a theoretical framework and a baseline computational survey of various nanotube-substrate configurations. The next step is to simulate more sophisticated interfaces to incorporate the effect of catalyst layers, oxides, imperfections, and uneven contact. Since chemical functionalization has been proposed as a means of bonding the nanotube to a substrate, thermal transport through molecular junctions is a natural extension of the modified mass nanotube tip research in this thesis.

A second major area of future work is in the theoretical and computational methods for estimating nanoscale thermal transport. At the nanotube-metal interface, the effect of electron-phonon processes is unclear and may play a significant role in the transport. In addition to electrical effects, quantum effects are important for phonon simulation below the Debye temperature. Future work may develop more robust computational tools to include these electrical and quantum effects on thermal transport.

APPENDIX A. NEMD TEMPERATURE PROFILES

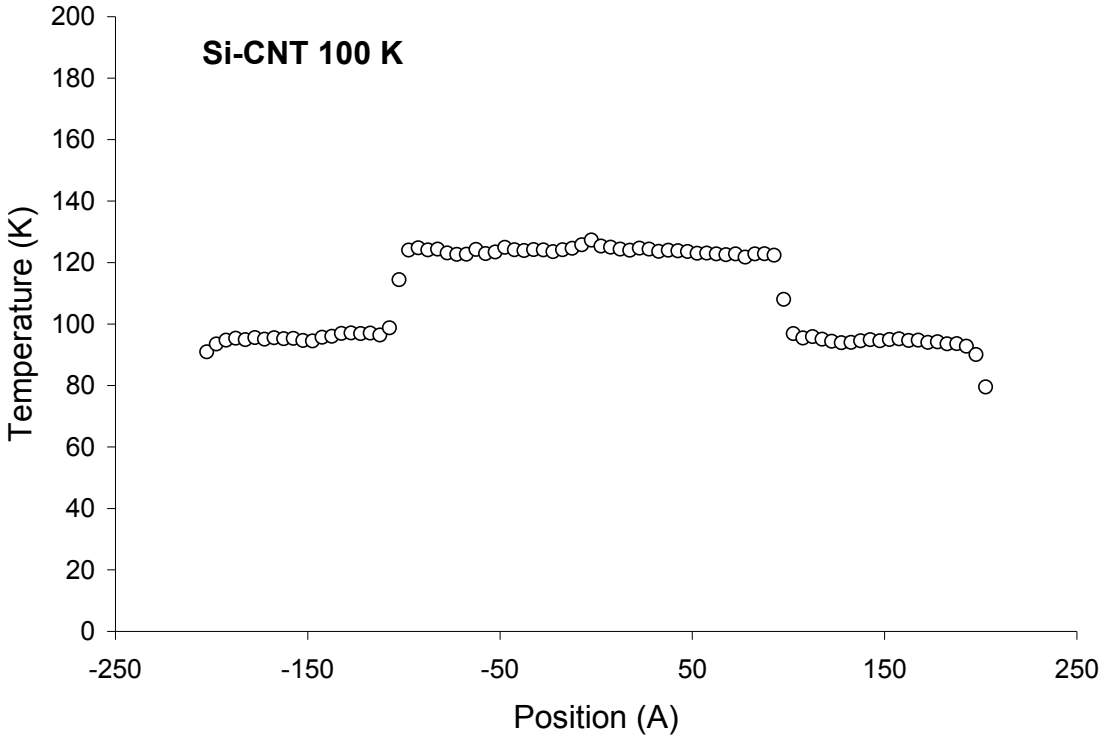

Figure A-1. Si-CNT Temperature Profile, 100K

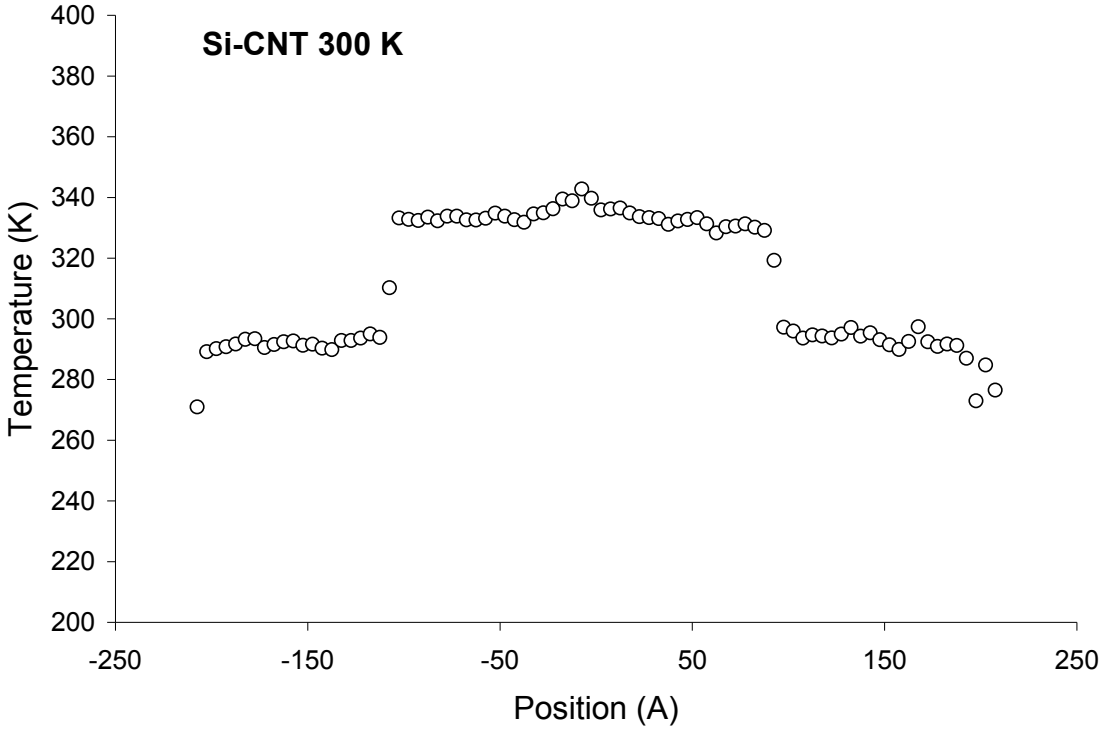

Figure A-2. Si-CNT Temperature Profile, 300K

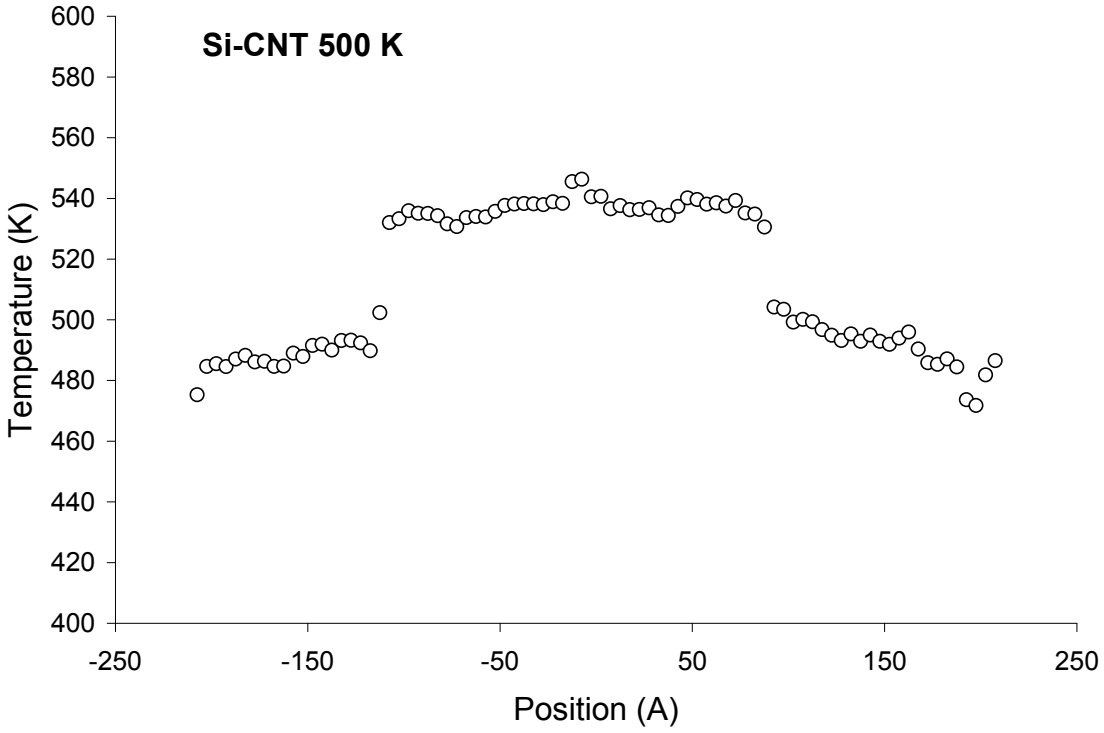

Figure A-3. Si-CNT Temperature Profile, 500K

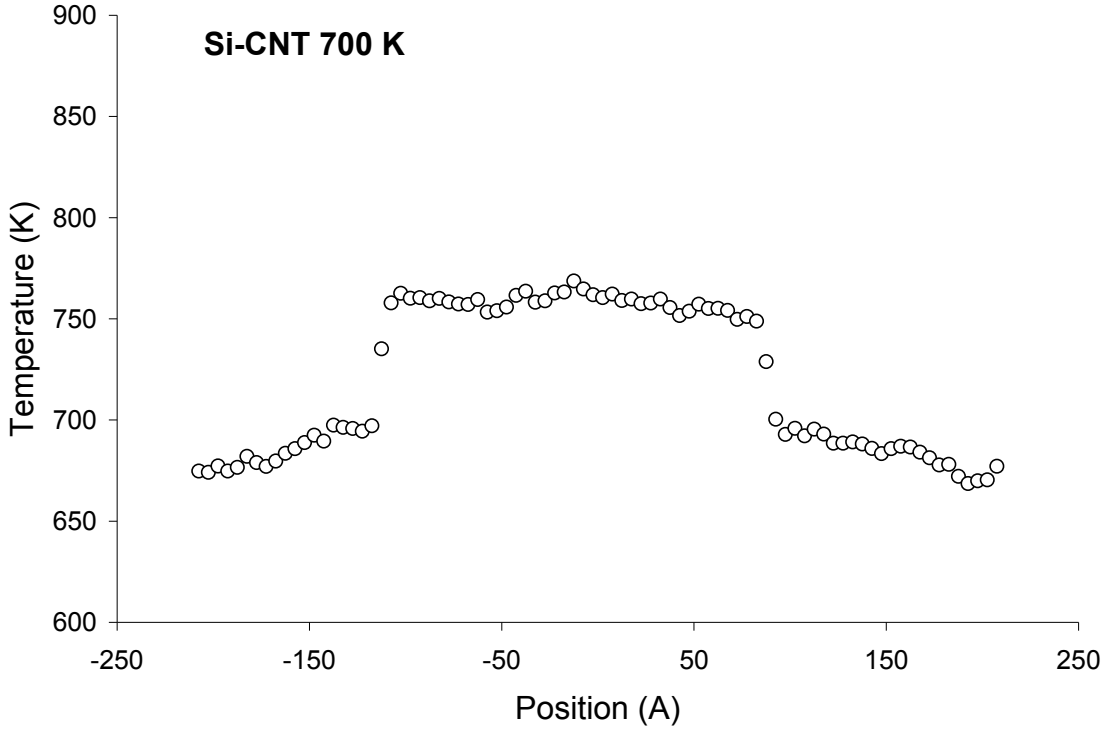

Figure A-4. Si-CNT Temperature Profile, 700K

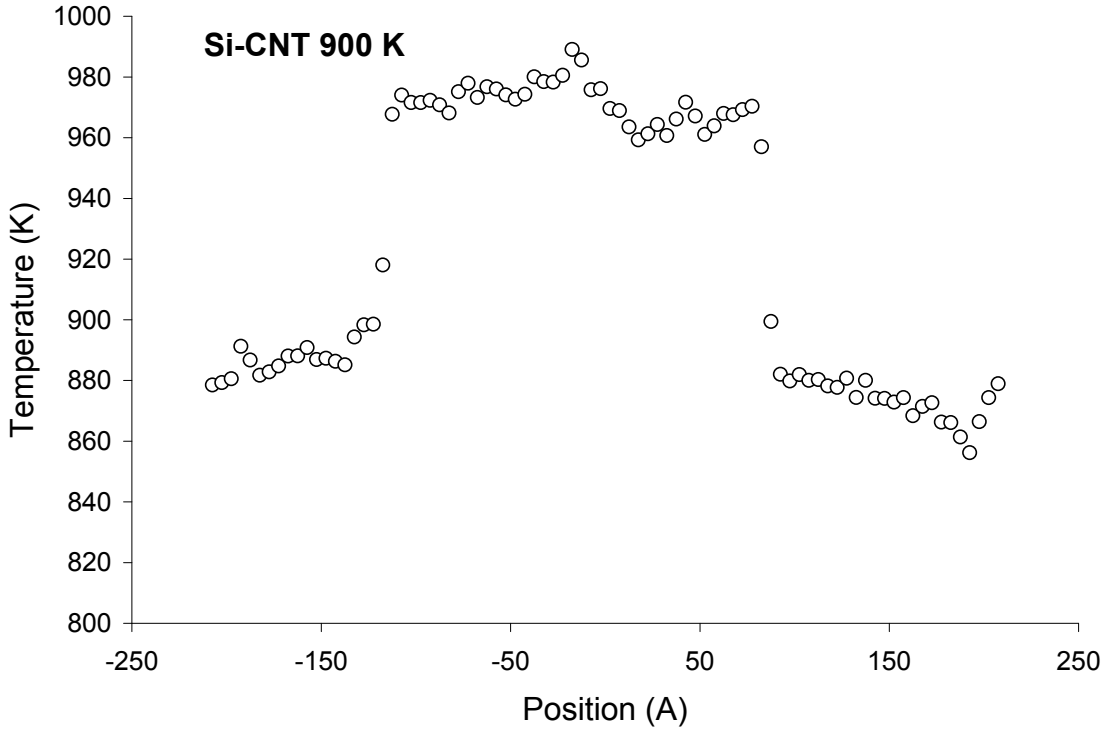

Figure A-5. Si-CNT Temperature Profile, 900K

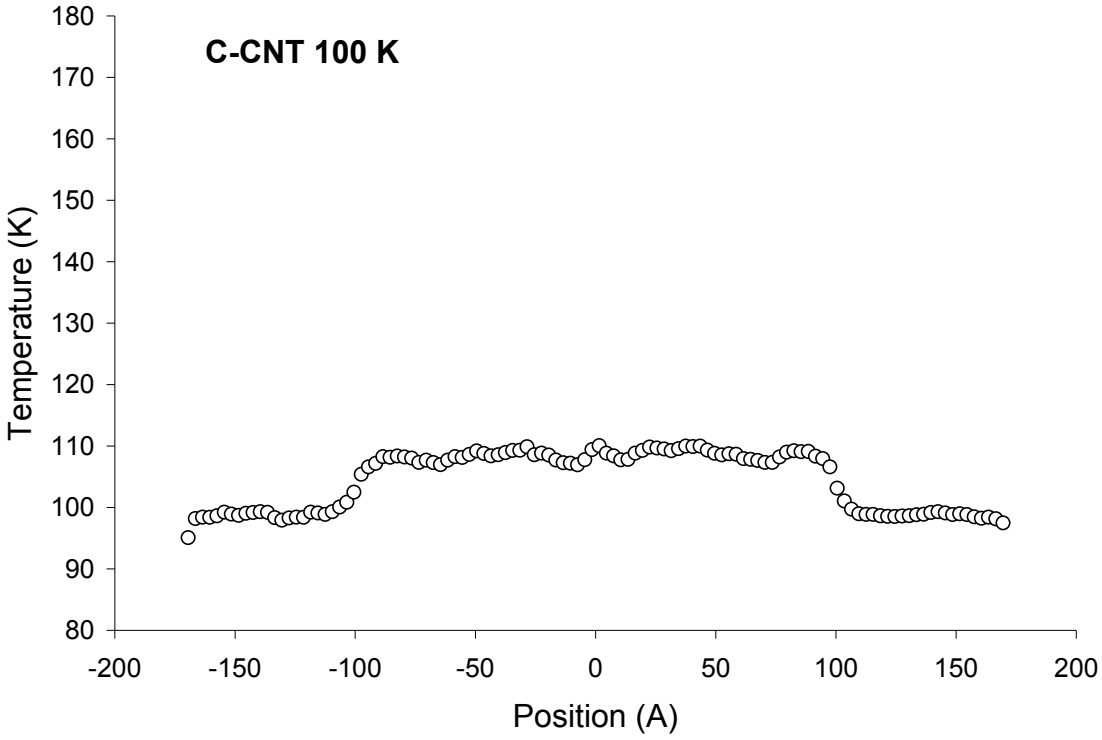

Figure A-6. C-CNT Temperature Profile, 100K

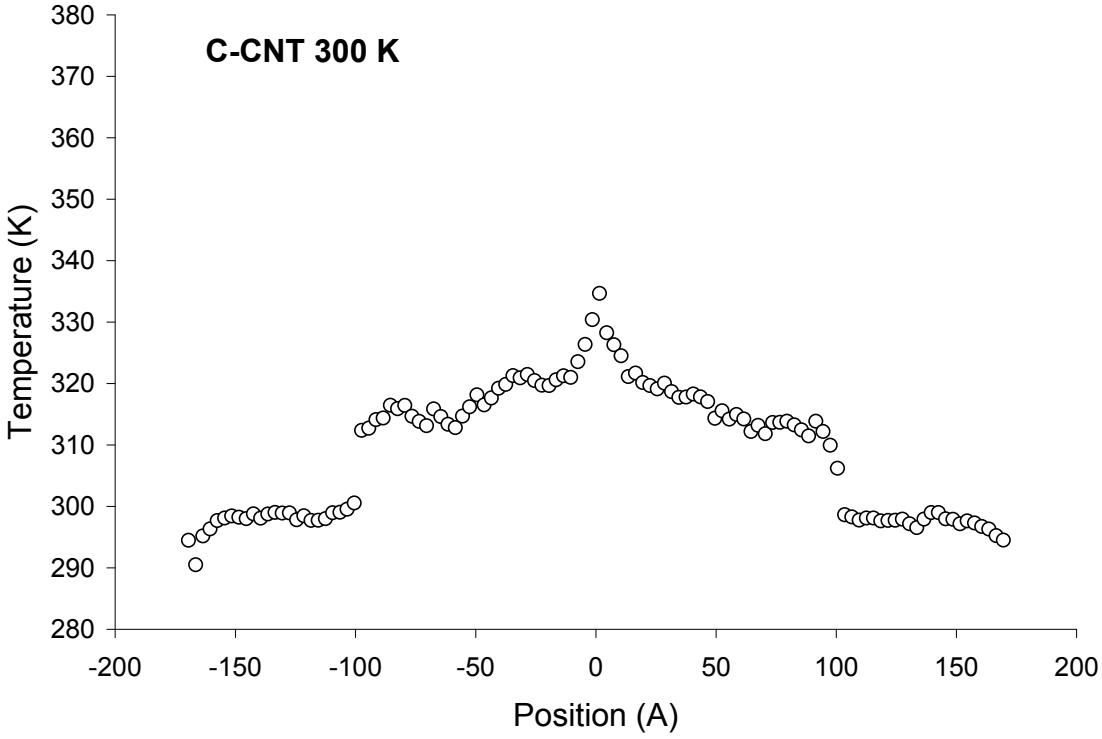

Figure A-7. C-CNT Temperature Profile, 300K

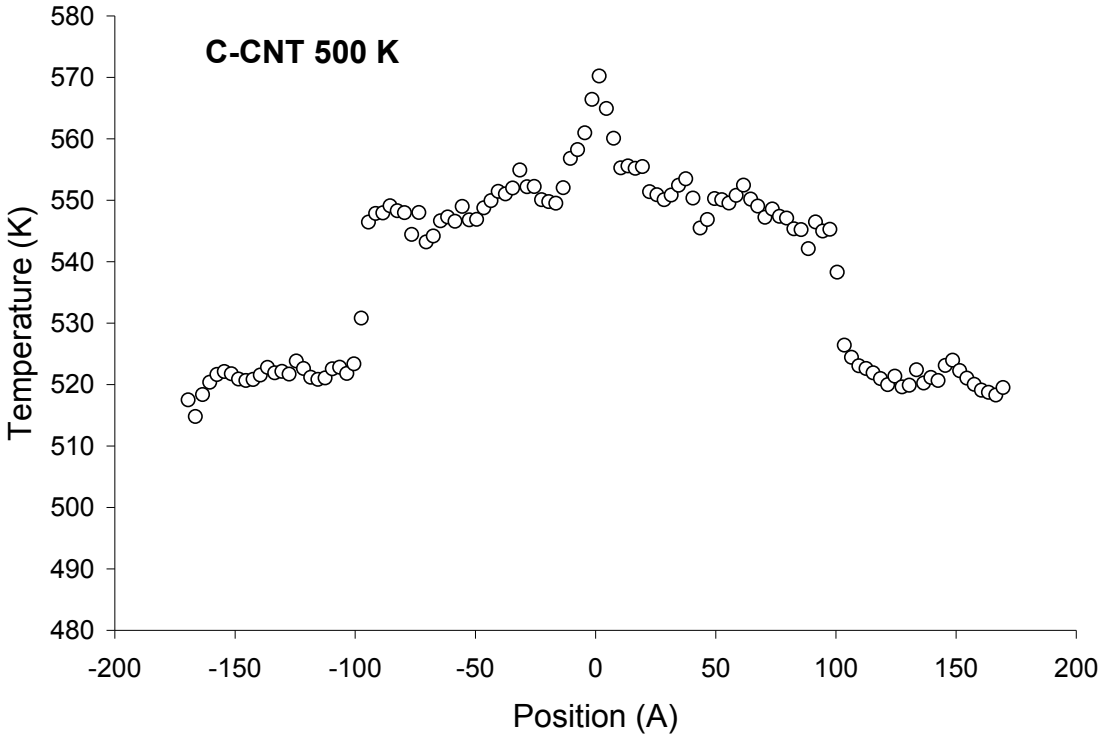

Figure A-8. C-CNT Temperature Profile, 500K

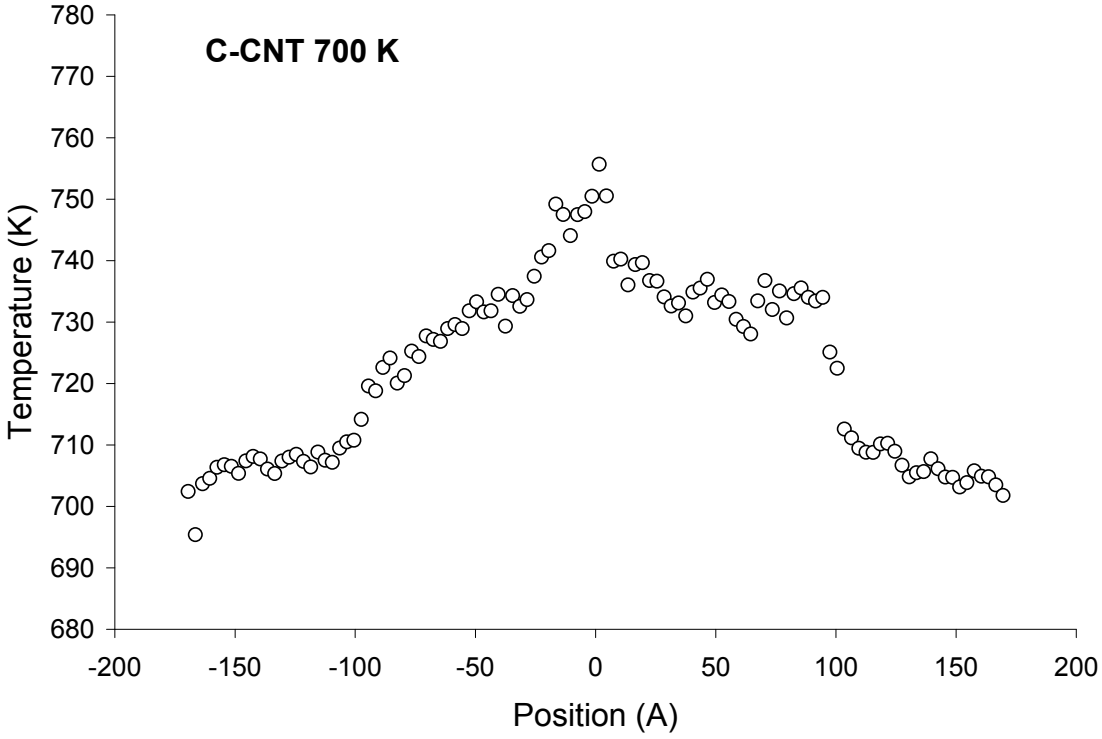

Figure A-9. C-CNT Temperature Profile, 700K

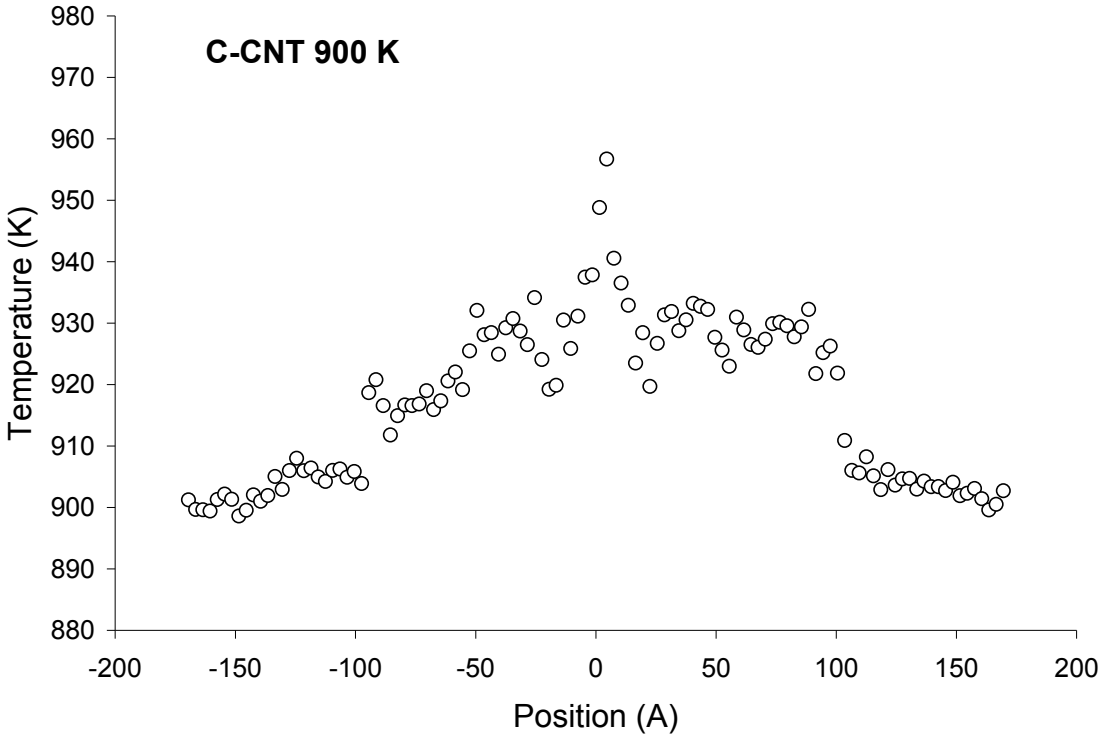

Figure A-10. C-CNT Temperature Profile, 900K

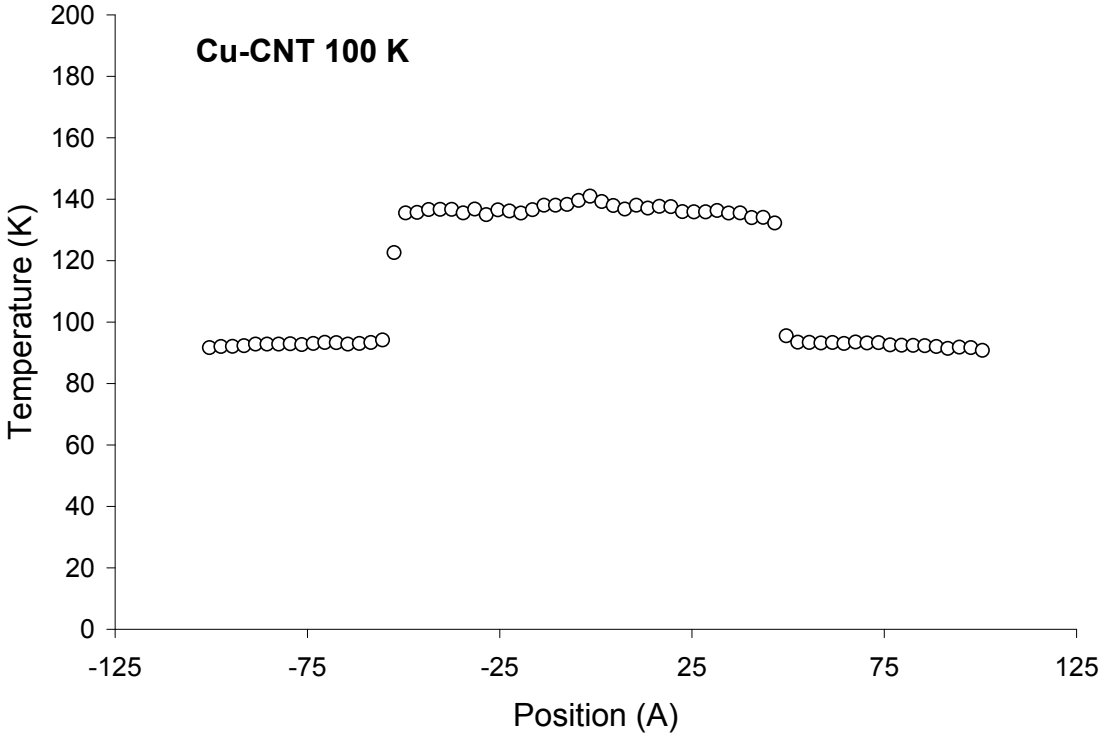

Figure A-11. Cu-CNT Temperature Profile, 100K

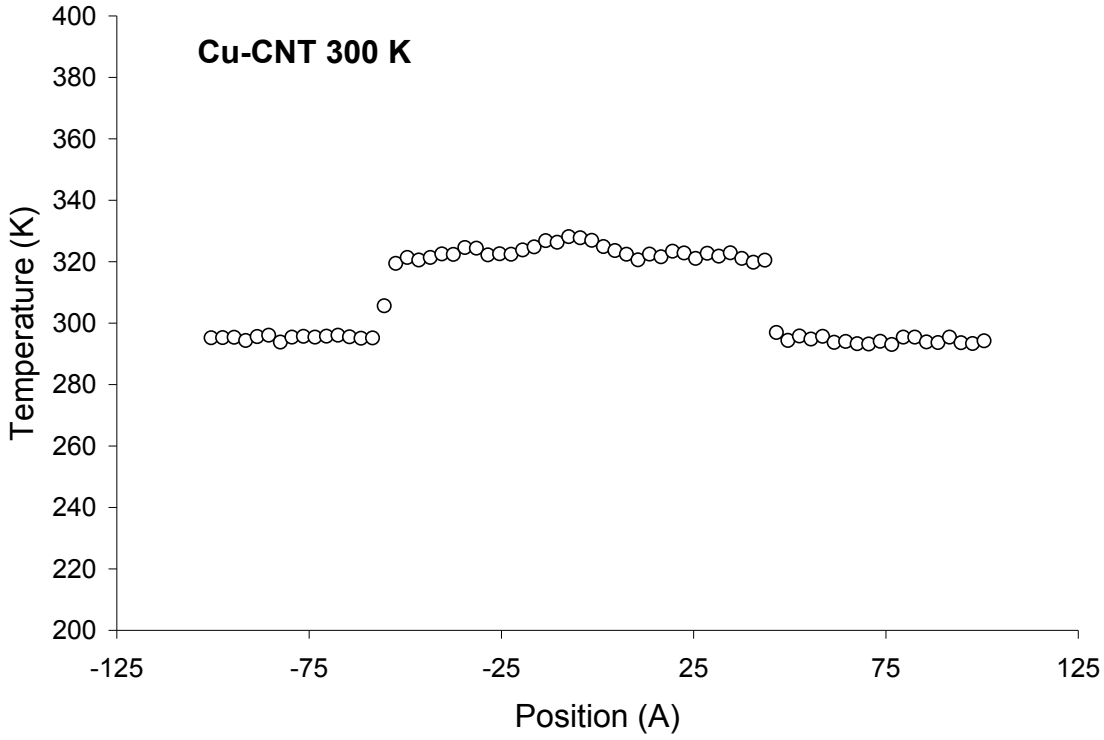

Figure A-12. Cu-CNT Temperature Profile, 300K

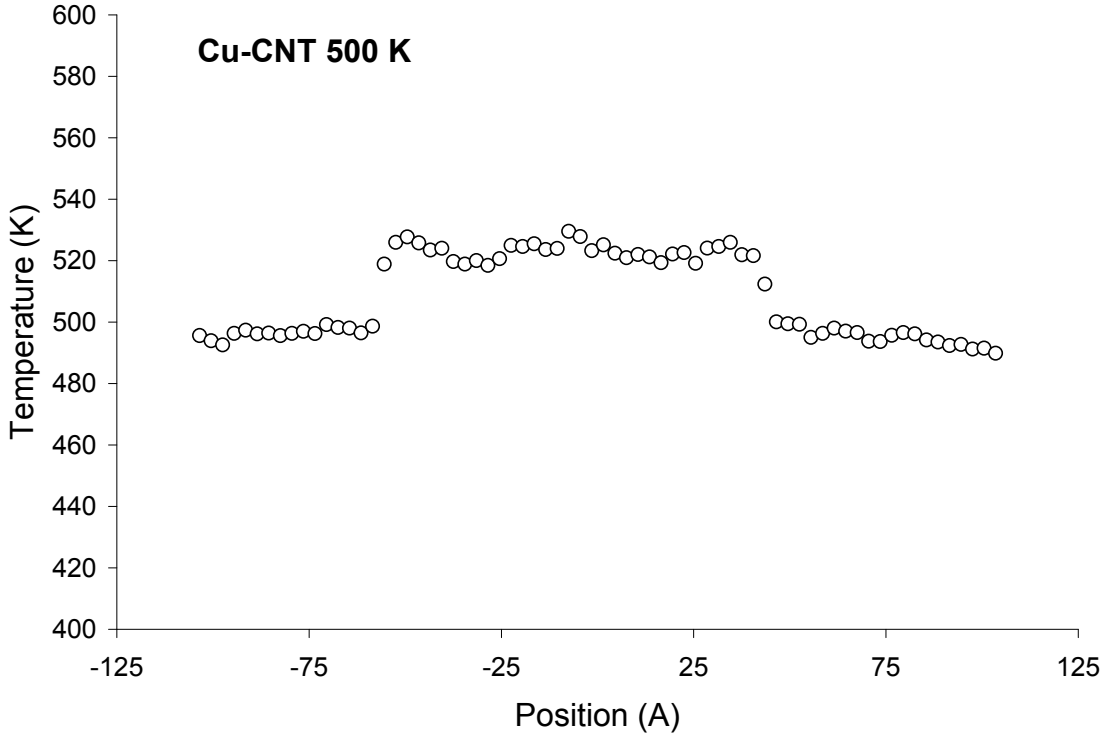

Figure A-13. Cu-CNT Temperature Profile, 500K

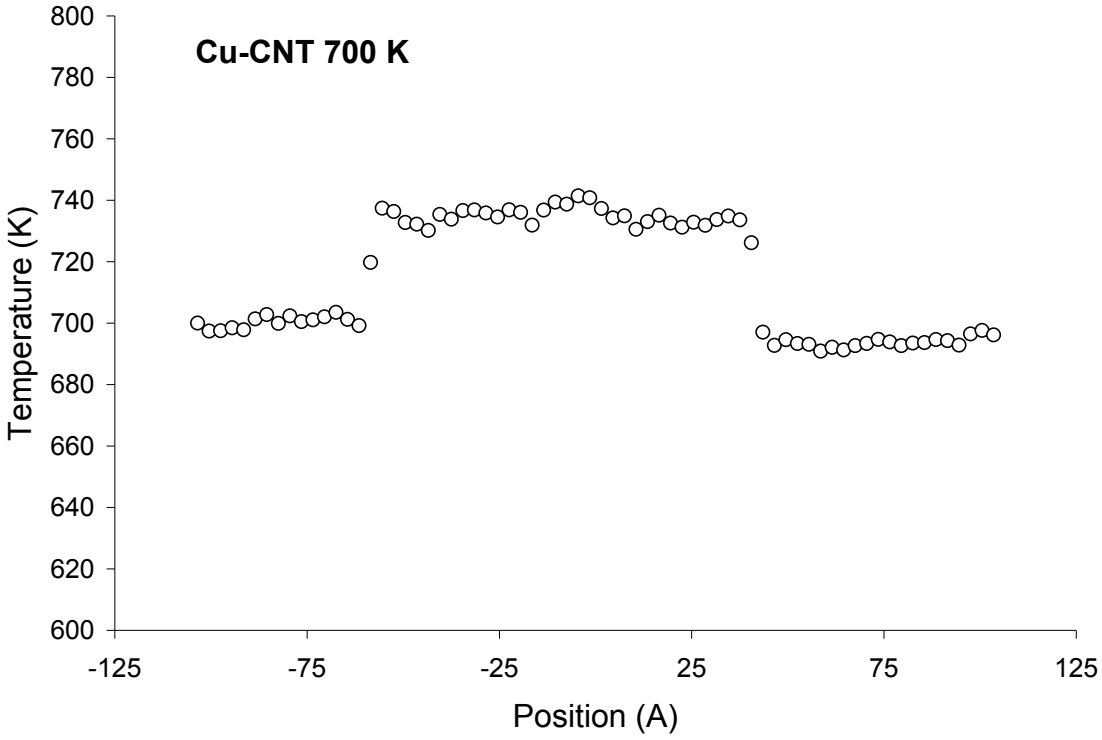

Figure A-14. Cu-CNT Temperature Profile, 700K

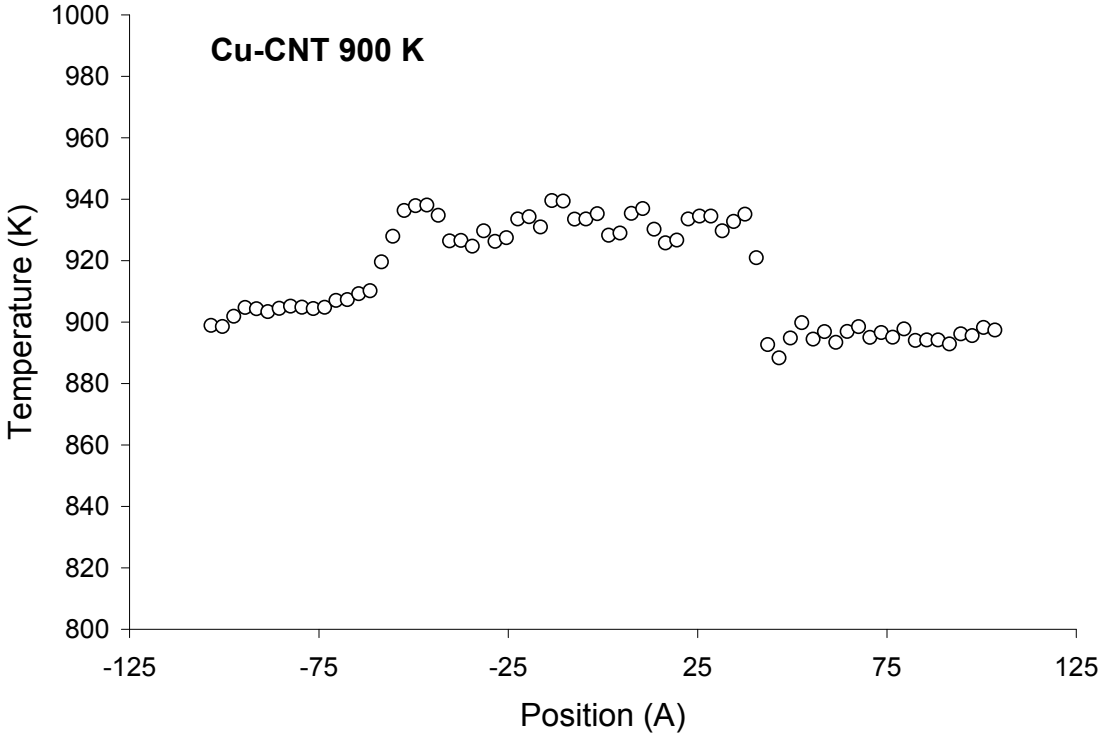

Figure A-15. Cu-CNT Temperature Profile, 900K

APPENDIX B. SAMPLE LAMMPS FILES

Appendix B contains the LAMMPS input files in.djr and SiC.brenner as well as the output file log.lammps for the large size silicon-CNT system at 300K.

in.djr

```
# CNT(10,10),200A,6x19x6_Si
units       metal
boundary    p p p

atom_style  atomic
read_data   data.djr

pair_style  brenner 1.0 0 0
pair_coeff  * * SiC.brenner Si C

mass        1 28.0855
mass        2 12.0107

neighbor    5.0 bin

thermo            10000
timestep    0.0005

group       qout id <> 1 72
group       qin  id <> 7125 7184
group       ASi  id <> 1 5544
group       BSi  id <> 8765 14236
group       cnt type 2
compute     mytemp all ke/atom

fix         1 all npt 300.0 300.0 1.0 aniso 0.0 0.0 0.0 0.0 0.0 0.0 1.0
drag 0.2
dump        4 all atom 10000 output.lammpstrj
run         50000
unfix       1

fix         2 all nve
fix         3 all ave/spatial 1 10000 10000 y lower 5.0 c_mytemp file
output.ke1
run         159040

dump        1 cnt custom 10 output.cnt tag x y z vx vy vz
dump        2 ASi custom 10 output.ASi tag x y z vx vy vz
dump        3 BSi custom 10 output.BSi tag x y z vx vy vz
run         40960

unfix       3
undump            1
undump            2
undump            3
```

```
fix         4 qin heat 1 3.0
fix         5 qout heat 1 -3.0
fix         6 all ave/spatial 1 10000 10000 y lower 5.0 c_mytemp file
output.ke2
run         300000
```

SiC.brenner

```
# Brenner potential parameters

1.39        R_CC
6.0         D_CC
2.1         beta_CC
1.22        S_CC
0.5         delta_CC
0.0         alpha_CC
1.7         rcmin_CC
2.0         rcmax_CC
0.00020813  a0_CC
108900            c02_CC
12.25       d02_CC
2.1970            R_SS
3.3870            D_SS
1.469       beta_SS
1.41        S_SS
0.78        delta_SS
0.0         alpha_SS
2.65        rcmin_SS
2.95        rcmax_SS
0.010       a0_SS
196         c02_SS
4.41        d02_SS
1.7631            R_CS
4.510       D_CS
1.698       beta_CS
1.492       S_CS
0.0         delta_CS
0.0         alpha_CS
2.2         rcmin_CS
2.5         rcmax_CS
0.0         a0_CS
0.0         c02_CS
0.0         d02_CS
```

log.lammps

```
LAMMPS (21 May 2008)
# CNT(10,10),200A,6x19x6_Si

units       metal
boundary    p p p

atom_style  atomic
read_data   data.djr
  orthogonal box = (-16.29 -203.25 -16.29) to (16.29 203.25 16.29)
  1 by 11 by 1 processor grid
  14236 atoms
  14236 velocities

pair_style  brenner 1.0 0 0
pair_coeff  * * SiC.brenner Si C

mass        1 28.0855
mass        2 12.0107

neighbor    5.0 bin

thermo            10000
timestep    0.0005

group       qout id <> 1 72
72 atoms in group qout
group       qin  id <> 7125 7184
60 atoms in group qin
group       ASi  id <> 1 5544
5544 atoms in group ASi
group       BSi  id <> 8765 14236
5472 atoms in group BSi
group       cnt type 2
3220 atoms in group cnt
compute     mytemp all ke/atom

fix         1 all npt 300.0 300.0 1.0 aniso 0.0 0.0 0.0 0.0 0.0 0.0 1.0
drag 0.2
dump        4 all atom 10000 output.lammpstrj
run         50000
Memory usage per processor = 8.382 Mbytes
Step Temp E_pair E_mol TotEng Press Volume
       0    300.02198   -74230.162            0   -73678.116
14723.778    431483.06
   10000     301.2516   -73961.342            0   -73407.034
795.53916    437287.06
   20000    318.07576   -73944.499            0   -73359.234   -
202.50593    438983.54
   30000     299.7739   -73965.148            0   -73413.558   -
484.73044    432930.03
   40000    298.12761   -73982.084            0   -73433.524
203.13639     437290.7
   50000    300.37821   -73979.575            0   -73426.873
187.76099    438119.58
```

```
Loop time of 2200.25 on 11 procs for 50000 steps with 14236 atoms

Pair  time (%) = 1120.45 (50.9238)
Neigh time (%) = 0.669042 (0.0304075)
Comm  time (%) = 684.362 (31.1038)
Outpt time (%) = 0.139319 (0.00633194)
Other time (%) = 394.63 (17.9357)

Nlocal:    1294.18 ave 1948 max 600 min
Histogram: 5 0 0 0 0 0 0 0 2 4
Nghost:    6085.45 ave 9521 max 2606 min
Histogram: 3 2 0 0 0 0 2 0 0 4
Neighs:    0 ave 0 max 0 min
Histogram: 11 0 0 0 0 0 0 0 0 0
FullNghs: 635772 ave 1.08998e+06 max 167750 min
Histogram: 5 0 0 0 0 0 0 1 1 4

Total # of neighbors = 6993490
Ave neighs/atom = 491.254
Neighbor list builds = 20
Dangerous builds = 0
unfix       1

fix         2 all nve
fix         3 all ave/spatial 1 10000 10000 y lower 5.0 c_mytemp file
output.ke1
run         159040
Memory usage per processor = 8.382 Mbytes
Step Temp E_pair E_mol TotEng Press
   50000    300.37821   -73979.575           0   -73426.873
187.76099
   60000    302.13043   -73982.799           0   -73426.873   -
363.38913
   70000    296.86575   -73973.101           0   -73426.863   -
1377.0553
   80000    299.90397   -73978.696           0   -73426.867   -
1145.1954
   90000     300.6035   -73979.988           0   -73426.872   -
2008.9593
  100000    300.29059   -73979.399           0   -73426.859   -
1629.4456
  110000    299.52146   -73977.988           0   -73426.863   -
2009.1482
  120000    297.41653   -73974.111           0    -73426.86    -
2299.556
  130000    297.23683   -73973.786           0   -73426.865   -
1692.0594
  140000    297.51526   -73974.302           0   -73426.869   -
2267.5035
  150000    299.00317   -73977.027           0   -73426.856   -
1617.1792
  160000    302.52558   -73983.508           0   -73426.856   -
1776.9944
  170000     301.7431   -73982.072           0   -73426.859   -
1884.6785
  180000    298.87175   -73976.776           0   -73426.847   -
1330.7657
```

```
  190000     300.2972   -73979.405            0   -73426.853   -
1821.5245
  200000    299.37377   -73977.686            0   -73426.832   -
973.79153
  209040    301.85418   -73982.263            0   -73426.846
233.30623
Loop time of 6852.09 on 11 procs for 159040 steps with 14236 atoms

Pair  time (%) = 3548.96 (51.7938)
Neigh time (%) = 0.168821 (0.00246379)
Comm  time (%) = 2197.92 (32.0766)
Outpt time (%) = 0.414114 (0.00604362)
Other time (%) = 1104.63 (16.1211)

Nlocal:    1294.18 ave 2016 max 584 min
Histogram: 5 0 0 0 0 0 1 0 1 4
Nghost:    6078.91 ave 9799 max 2415 min
Histogram: 4 1 0 0 0 0 2 0 0 4
Neighs:    0 ave 0 max 0 min
Histogram: 11 0 0 0 0 0 0 0 0 0
FullNghs: 634407 ave 1.13876e+06 max 163124 min
Histogram: 5 0 0 0 0 0 1 1 0 4

Total # of neighbors = 6978476
Ave neighs/atom = 490.199
Neighbor list builds = 5
Dangerous builds = 0

dump        1 cnt custom 10 output.cnt tag x y z vx vy vz
dump        2 ASi custom 10 output.ASi tag x y z vx vy vz
dump        3 BSi custom 10 output.BSi tag x y z vx vy vz
run         40960
Memory usage per processor = 9.60593 Mbytes
Step Temp E_pair E_mol TotEng Press
  209040    301.85418   -73982.263            0   -73426.846
233.30623
  210000    301.67882   -73981.949            0   -73426.854   -
993.94908
  220000    302.63857   -73983.713            0   -73426.853   -
630.85765
  230000    300.38835   -73979.569            0   -73426.849
221.69696
  240000    302.02898     -73982.6            0   -73426.861   -
1.8211686
  250000    300.73671   -73980.203            0   -73426.842
898.23899
Loop time of 2082.26 on 11 procs for 40960 steps with 14236 atoms

Pair  time (%) = 929.217 (44.6255)
Neigh time (%) = 0 (0)
Comm  time (%) = 591.388 (28.4013)
Outpt time (%) = 216.556 (10.4001)
Other time (%) = 345.096 (16.5732)

Nlocal:    1294.18 ave 2012 max 580 min
Histogram: 5 0 0 0 0 0 1 0 1 4
Nghost:    6143.91 ave 10136 max 2400 min
```

```
Histogram: 4 1 0 0 0 1 1 0 0 4
Neighs:    0 ave 0 max 0 min
Histogram: 11 0 0 0 0 0 0 0 0 0
FullNghs: 636578 ave 1.13648e+06 max 161486 min
Histogram: 5 0 0 0 0 0 1 0 1 4

Total # of neighbors = 7002354
Ave neighs/atom = 491.877
Neighbor list builds = 0
Dangerous builds = 0

unfix       3
undump            1
undump            2
undump            3

fix         4 qin heat 1 3.0
fix         5 qout heat 1 -3.0
fix         6 all ave/spatial 1 10000 10000 y lower 5.0 c_mytemp file
output.ke2
run         300000
Memory usage per processor = 8.54544 Mbytes
Step Temp E_pair E_mol TotEng Press
  250000    300.73671   -73980.203            0   -73426.842
898.23899
  260000    301.08082   -73980.832            0   -73426.838
461.43973
  270000    302.05344   -73982.608            0   -73426.824
922.1894
  280000    302.88904   -73984.155            0   -73426.834
1009.4042
  290000    299.34739   -73977.652            0   -73426.848
527.71643
  300000    299.53978   -73978.005            0   -73426.846
1029.8747
  310000     304.7659   -73987.621            0   -73426.846   -
133.80102
  320000    301.92903   -73982.391            0   -73426.836
141.83151
  330000    303.23605   -73984.808            0   -73426.848   -
403.65689
  340000    303.28432   -73984.907            0   -73426.858   -
793.05931
  350000    301.39952   -73981.427            0   -73426.846   -
377.97689
  360000    301.77573   -73982.123            0   -73426.851   -
1255.6701
  370000    300.58196   -73979.912            0   -73426.836   -
900.06628
  380000    302.64444   -73983.704            0   -73426.832   -
1294.7349
  390000    303.09459    -73984.51            0    -73426.81   -
1334.4308
  400000    299.96795   -73978.754            0   -73426.807   -
1539.6211
  410000    300.12505   -73979.027            0   -73426.792   -
2121.8763
```

```
  420000    297.66766   -73974.508            0   -73426.794   -
1391.3723
  430000    297.96011    -73975.06            0   -73426.808   -
2187.9969
  440000    300.79672   -73980.285            0   -73426.814   -
1759.3554
  450000    300.12843   -73979.058            0   -73426.817   -
1756.7085
  460000    298.40554   -73975.892            0    -73426.82   -
1853.8445
  470000    301.24736   -73981.113            0   -73426.813   -
1203.5033
  480000    301.18355   -73981.016            0   -73426.833   -
1288.8213
  490000    301.48281   -73981.561            0   -73426.827   -
124.24568
  500000    301.89179   -73982.293            0   -73426.806   -
382.07213
  510000    303.38158   -73985.041            0   -73426.813   -
100.30013
  520000    301.35208   -73981.296            0   -73426.802
232.42791
  530000    303.27467   -73984.833            0   -73426.802
35.661623
  540000    300.79668   -73980.268            0   -73426.797
555.59636
  550000    301.42334   -73981.421            0   -73426.796
279.03487
Loop time of 13634.2 on 11 procs for 300000 steps with 14236 atoms

Pair  time (%) = 6793.35 (49.8257)
Neigh time (%) = 0.202439 (0.00148478)
Comm  time (%) = 4404.29 (32.3032)
Outpt time (%) = 0.836708 (0.00613682)
Other time (%) = 2435.54 (17.8635)

Nlocal:    1294.18 ave 2016 max 582 min
Histogram: 5 0 0 0 0 1 0 0 0 5
Nghost:    6358.27 ave 10009 max 2655 min
Histogram: 4 0 1 0 0 1 0 1 0 4
Neighs:    0 ave 0 max 0 min
Histogram: 11 0 0 0 0 0 0 0 0 0
FullNghs: 637066 ave 1.14374e+06 max 162046 min
Histogram: 5 0 0 0 1 0 0 0 1 4

Total # of neighbors = 7007728
Ave neighs/atom = 492.254
Neighbor list builds = 6
Dangerous builds = 0
```

APPENDIX C. POST-PROCESSING CODES

Appendix C contains the codes Dispersion.f90, DOS.f90, and Kinetic.f90 used to calculate the dispersion relations, phonon density of states, and temperature profiles, respectively.

Dispersion.f90

```
!  Dispersion.f90
    program Dispersion
    implicit none
    real*8,allocatable :: v(:,:,:)
    integer steps,natoms,cell,ncells,tag,frames,unit
    integer i,j,k,m,n,t,zeta
    real*8 vy,vx,vz,x,y,z

    steps=4096
    natoms=1000
    unit=200
    ncells=5 !Change in format


    open(unit=100,file='output.BSi1',status='old')
    allocate(v(unit,ncells,steps))

    do t=1,steps
       do i=1,9               !Skip first 9 lines
         read(100,*)
       enddo
       do j=1,natoms
         read(100,*) tag,x,y,z,vx,vy,vz
         do zeta=1,ncells
            if ((tag-1830).le.(unit*zeta)) goto 10
          enddo
10        n=tag-unit*(zeta-1)-1830
           v(n,zeta,t)=vy
!          print *,t,tag-1830,zeta,n,vy
       enddo
      print *,t
    enddo

    open(unit=500,file='disp.BSi1',status='replace')
    do n=1,unit
!      write(500,*)n
      do t=1,steps
        write(500,17) (v(n,zeta,t),zeta=1,ncells)
17  format(5F10.5)
      enddo
    enddo
    close(500)

    end program Dispersion
```

DOS.f90

```
!  DOS.f90

    program DOS

    implicit none

    integer steps,natoms,i,j,k,m
    real*8 vx,vy,vz,x,y,z,tag

    real*8,allocatable :: vacf(:),v(:,:,:)

    steps=2048
    natoms=5472

    open(unit=100,file='output.cnt',status='old')

    allocate(vacf(steps-1))
    allocate(v(steps,natoms,3))

    natoms=3220

    do k=1,steps
        read(100,*)
        read(100,*)
        read(100,*)
        read(100,*)
        read(100,*)
        read(100,*)
        read(100,*)
        read(100,*)
        read(100,*)

   !Read the History File
        do i=1,natoms
            read(100,*) tag,x,y,z,vx,vy,vz
            v(k,i,1)=vx
            v(k,i,2)=vy
            v(k,i,3)=vz
        end do
    enddo

    !Velocity Autocorrelation Function
    do j=1,steps-1
        vacf(j)=0.0d0
    enddo

    print *,'Calculating VACF...  cnt'

    do k=1,steps-1
        do j=k+1,steps
            do i=1,natoms
            vacf(j-k)=vacf(j-
k)+v(k,i,1)*v(j,i,1)+v(k,i,2)*v(j,i,2)+v(k,i,3)*v(j,i,3)
!            vacf(j-k)=vacf(j-k)+v(k,i,2)*v(j,i,2)
            end do
```

```
        end do
!    print *,k
    end do

    do j=1,steps-1
!    print *,j
    vacf(j)=vacf(j)/((steps-j)*natoms)
    enddo

    open(unit=500,file='VACF.cnt',status='replace')
17  format(f10.5,x,f20.3)
    do i=1,steps-1
    write(500,17)i*0.001,vacf(i)
    enddo
    close(500)
    print *,'Finished CNT'

!!!!!!!!!!!!!!!

    open(unit=103,file='output.BSi',status='old')

    natoms=5472

    do k=1,steps
        read(103,*)
        read(103,*)
        read(103,*)
        read(103,*)
        read(103,*)
        read(103,*)
        read(103,*)
        read(103,*)
        read(103,*)

   !Read the History File
        do i=1,natoms
            read(103,*) tag,x,y,z,vx,vy,vz
            v(k,i,1)=vx
            v(k,i,2)=vy
            v(k,i,3)=vz
        end do
    enddo

    !Velocity Autocorrelation Function
    do j=1,steps-1
        vacf(j)=0.0d0
    enddo

    print *,'Calculating VACF...  BSi'

    do k=1,steps-1
        do j=k+1,steps
            do i=1,natoms
            vacf(j-k)=vacf(j-
k)+v(k,i,1)*v(j,i,1)+v(k,i,2)*v(j,i,2)+v(k,i,3)*v(j,i,3)
!            vacf(j-k)=vacf(j-k)+v(k,i,2)*v(j,i,2)
            end do
```

```
        end do
!     print *,k
      end do

      do j=1,steps-1
!     print *,j
      vacf(j)=vacf(j)/((steps-j)*natoms)
      enddo

      open(unit=503,file='VACF.BSi',status='replace')

      do i=1,steps-1
      write(503,17)i*0.001,vacf(i)
      enddo
      close(503)

      end program DOS
```

Kinetic.f90

```
!  Kinetic.f90

    program Kinetic
    implicit none

    integer slabs,id
    integer i,j,k,rec,c1,c2,c3
    real*8 ke,num,count,pos,sum1,sum2,sum3,drop
    real*8,allocatable :: temp(:,:),x(:)
    slabs=84
    rec=30
    allocate(temp(rec,slabs))
    allocate(x(slabs))

    open(unit=101,file='output.ke2',status='old')

    do i=1,3                  !Skip first 3 lines
      read(101,*)
    enddo

    do j=1,rec
    read(101,*)
    count=0
    do i=1,slabs
      read(101,*) id,pos,num,ke
       temp(j,i)=(2*11604.50501*ke)/3
       if(j.eq.1)then
       x(i)=pos
       endif
       count=count+num
!     print *,i,pos,ke,temp(j,i),count
    enddo
    enddo
    close(101)

sum3/c3)/2,(sum2/c2+sum1/c1+sum3/c3)/3

    open(unit=501,file='output.temp2',status='replace')
        write(501,17) (x(i),i=1,slabs)
    do j=1,rec
        write(501,17) (temp(j,i),i=1,slabs)
17  format(84F15.6)
      enddo

    close(501)

   end program Kinetic
```

APPENDIX D. BRENNER POTENTIAL IMPLEMENTATION

pair_brenner.h

```
/* ----------------------------------------------------------------------
   LAMMPS - Large-scale Atomic/Molecular Massively Parallel Simulator
   http://lammps.sandia.gov, Sandia National Laboratories
   Steve Plimpton, sjplimp@sandia.gov

   Copyright (2003) Sandia Corporation.  Under the terms of Contract
   DE-AC04-94AL85000 with Sandia Corporation, the U.S. Government retains
   certain rights in this software.  This software is distributed under
   the GNU General Public License.

   See the README file in the top-level LAMMPS directory.
------------------------------------------------------------------------- */

#ifndef PAIR_BRENNER_H
#define PAIR_BRENNER_H

#include "pair.h"

namespace LAMMPS_NS {

class PairBRENNER : public Pair {
 public:
  PairBRENNER(class LAMMPS *);
  ~PairBRENNER();
  void compute(int, int);
  void settings(int, char **);
  void coeff(int, char **);
  void init_style();
  double init_one(int, int);
  double memory_usage();

 private:
  int me;
  int ljflag,torflag;              // 0/1 if LJ,torsion terms included
  int maxlocal;                    // size of numneigh, firstneigh arrays
  int **pages;                     // neighbor list pages
  int maxpage;                     // # of pages currently allocated
  int pgsize;                      // size of neighbor page
  int oneatom;                     // max # of neighbors for one atom
  int npage;                       // current page in page list
  int *map;                        // 0 (C), 1 (H), or -1 (NULL) for each type
  double cutlj;                    // user-specified LJ cutoff
  double cutljrebosq;              // cut for when to compute
                                   // REBO neighs of ghost atoms

  double **cutljsq;                // LJ cutoffs for C,H types
  double **lj1,**lj2,**lj3,**lj4;  // pre-computed LJ coeffs for C,H types
  double cut3rebo;                 // maximum distance for 3rd REBO neigh

  int *REBO_numneigh;              // # of pair neighbors for each atom
  int **REBO_firstneigh;           // ptr to 1st neighbor of each atom
  double *nC,*nH;                  // sum of weighting fns with REBO neighs

  double smin,Nmin,Nmax,NCmin,NCmax,thmin,thmax;
  double rcmin[2][2],rcmax[2][2],rcmaxsq[2][2],rcmaxp[2][2];
  double Q[2][2],alpha[2][2],A[2][2],rho[2][2],BIJc[2][2][3],Beta[2][2][3];
```

```
  double rcLJmin[2][2],rcLJmax[2][2],rcLJmaxsq[2][2],bLJmin[2][2],bLJmax[2][2];
  double epsilon[2][2],sigma[2][2],epsilonT[2][2];

  // Brenner Parameters

  double R[2][2],D[2][2],bet[2][2],S[2][2],del[2][2],alp[2][2];
  double rmin[2][2],rmax[2][2],a[2][2],c[2][2],d[2][2];

  // spline coefficients

  double gCdom[5],gC1[4][6],gC2[4][6],gHdom[4],gH[3][6];
  double pCCdom[2][2],pCHdom[2][2],pCC[4][4][16],pCH[4][4][16];
  double piCCdom[3][2],piCHdom[3][2],piHHdom[3][2];
  double piCC[4][4][9][64],piCH[4][4][9][64],piHH[4][4][9][64];
  double Tijdom[3][2],Tijc[4][4][9][64];

  // spline knot values

  double PCCf[5][5],PCCdfdx[5][5],PCCdfdy[5][5],PCHf[5][5];
  double PCHdfdx[5][5],PCHdfdy[5][5];
  double piCCf[5][5][10],piCCdfdx[5][5][10];
  double piCCdfdy[5][5][10],piCCdfdz[5][5][10];
  double piCHf[5][5][10],piCHdfdx[5][5][10];
  double piCHdfdy[5][5][10],piCHdfdz[5][5][10];
  double piHHf[5][5][10],piHHdfdx[5][5][10];
  double piHHdfdy[5][5][10],piHHdfdz[5][5][10];
  double Tf[5][5][10],Tdfdx[5][5][10],Tdfdy[5][5][10],Tdfdz[5][5][10];

  void REBO_neigh();
  void FREBO(int, int);
  void FLJ(int, int);
  void TORSION(int, int);

  double bondorder(int, int, double *, double, double, double **, int);
  double bondorderLJ(int, int, double *, double, double,
                 double *, double, double **, int);

  double Sp(double, double, double, double &);
  double Sp2(double, double, double, double &);

  double gSpline(double, double, int, double *, double *);
  double PijSpline(double, double, int, int, double *);
  double piRCSpline(double, double, double, int, int, double *);
  double TijSpline(double, double, double, double *);

  double kronecker(int, int);

  void add_pages(int);
  void read_file(char *);

  double Sp5th(double, double *, double *);
  double Spbicubic(double, double, double *, double *);
  double Sptricubic(double, double, double, double *, double *);
  void spline_init();

  void allocate();
};

}

#endif
```

pair_brenner.cpp

```
/* ----------------------------------------------------------------------
   LAMMPS - Large-scale Atomic/Molecular Massively Parallel Simulator
   http://lammps.sandia.gov, Sandia National Laboratories
   Steve Plimpton, sjplimp@sandia.gov

   Copyright (2003) Sandia Corporation.  Under the terms of Contract
   DE-AC04-94AL85000 with Sandia Corporation, the U.S. Government retains
   certain rights in this software.  This software is distributed under
   the GNU General Public License.

   See the README file in the top-level LAMMPS directory.
------------------------------------------------------------------------- */

/* ----------------------------------------------------------------------
   Contributing author: D. Rogers 03Feb09
------------------------------------------------------------------------- */

#include "math.h"
#include "stdio.h"
#include "stdlib.h"
#include "string.h"
#include "mpi.h"
#include "pair_brenner.h"
#include "atom.h"
#include "neighbor.h"
#include "neigh_request.h"
#include "force.h"
#include "comm.h"
#include "neighbor.h"
#include "neigh_list.h"
#include "neigh_request.h"
#include "memory.h"
#include "error.h"

using namespace LAMMPS_NS;

#define MIN(a,b) ((a) < (b) ? (a) : (b))
#define MAX(a,b) ((a) > (b) ? (a) : (b))

#define MAXLINE 1024
#define TOL 1.0e-9
#define PI 3.14159265
#define PGDELTA 1

/* ---------------------------------------------------------------------- */

PairBRENNER::PairBRENNER(LAMMPS *lmp) : Pair(lmp)
{
  single_enable = 0;
  one_coeff = 1;

  maxlocal = 0;
  REBO_numneigh = NULL;
  REBO_firstneigh = NULL;
  maxpage = 0;
  pages = NULL;
  nC = nH = NULL;
}
```

```
/* ----------------------------------------------------------------------
   check if allocated, since class can be destructed when incomplete
------------------------------------------------------------------------- */

PairBRENNER::~PairBRENNER()
{
  memory->sfree(REBO_numneigh);
  memory->sfree(REBO_firstneigh);
  for (int i = 0; i < maxpage; i++) memory->sfree(pages[i]);
  memory->sfree(pages);
  memory->sfree(nC);
  memory->sfree(nH);

  if (allocated) {
    memory->destroy_2d_int_array(setflag);
    memory->destroy_2d_double_array(cutsq);

    memory->destroy_2d_double_array(cutljsq);
    memory->destroy_2d_double_array(lj1);
    memory->destroy_2d_double_array(lj2);
    memory->destroy_2d_double_array(lj3);
    memory->destroy_2d_double_array(lj4);
    delete [] map;
  }
}

/* ----------------------------------------------------------------------- */

void PairBRENNER::compute(int eflag, int vflag)
{
  if (eflag || vflag) ev_setup(eflag,vflag);
  else evflag = vflag_fdotr = 0;

  REBO_neigh();
  FREBO(eflag,vflag);
  if (vflag_fdotr) virial_compute();
}

/* ----------------------------------------------------------------------
   allocate all arrays
------------------------------------------------------------------------- */

void PairBRENNER::allocate()
{
  allocated = 1;
  int n = atom->ntypes;

  setflag = memory->create_2d_int_array(n+1,n+1,"pair:setflag");
  for (int i = 1; i <= n; i++)
    for (int j = i; j <= n; j++)
      setflag[i][j] = 0;

  cutsq = memory->create_2d_double_array(n+1,n+1,"pair:cutsq");
  map = new int[n+1];
}

/* ----------------------------------------------------------------------
   global settings
------------------------------------------------------------------------- */

void PairBRENNER::settings(int narg, char **arg)
{
  if (narg != 1 && narg != 3) error->all("Illegal pair_style command");
```

```
  cutlj = atof(arg[0]);

  ljflag = torflag = 1;
  if (narg == 3) {
    ljflag = atoi(arg[1]);
    torflag = atoi(arg[2]);
  }
}

/* ----------------------------------------------------------------------
   set coeffs for one or more type pairs
------------------------------------------------------------------------- */

void PairBRENNER::coeff(int narg, char **arg)
{
  if (!allocated) allocate();

  if (narg != 3 + atom->ntypes)
    error->all("Incorrect args for pair coefficients");

  // insure I,J args are * *

  if (strcmp(arg[0],"*") != 0 || strcmp(arg[1],"*") != 0)
    error->all("Incorrect args for pair coefficients");

  // read args that map atom types to C and H
  // map[i] = which element (0,1) the Ith atom type is, -1 if NULL

  for (int i = 3; i < narg; i++) {
    if (strcmp(arg[i],"NULL") == 0) {
      map[i-2] = -1;
      continue;
    } else if (strcmp(arg[i],"C") == 0) {
      map[i-2] = 0;
    } else if (strcmp(arg[i],"Si") == 0) {
      map[i-2] = 1;
    } else error->all("Incorrect args for pair coefficients");
  }

  // read potential file and initialize fitting splines

  read_file(arg[2]);
  // spline_init();

  // clear setflag since coeff() called once with I,J = * *

  int n = atom->ntypes;
  for (int i = 1; i <= n; i++)
    for (int j = i; j <= n; j++)
      setflag[i][j] = 0;

  // set setflag i,j for type pairs where both are mapped to elements

  int count = 0;
  for (int i = 1; i <= n; i++)
    for (int j = i; j <= n; j++)
      if (map[i] >= 0 && map[j] >= 0) {
       setflag[i][j] = 1;
       count++;
      }

  if (count == 0) error->all("Incorrect args for pair coefficients");
```

```
}

/* ----------------------------------------------------------------------
   init specific to this pair style
------------------------------------------------------------------------- */

void PairBRENNER::init_style()
{
  if (atom->tag_enable == 0)
    error->all("Pair style Brenner requires atom IDs");
  if (force->newton_pair == 0)
    error->all("Pair style Brenner requires newton pair on");

  // need a full neighbor list

  int irequest = neighbor->request(this);
  neighbor->requests[irequest]->half = 0;
  neighbor->requests[irequest]->full = 1;

  // local REBO neighbor list memory

  pgsize = neighbor->pgsize;
  oneatom = neighbor->oneatom;
  if (maxlocal == 0) add_pages(0);
}

/* ----------------------------------------------------------------------
   init for one type pair i,j and corresponding j,i
------------------------------------------------------------------------- */

double PairBRENNER::init_one(int i, int j)
{
  if (setflag[i][j] == 0) error->all("All pair coeffs are not set");

  // convert to C,H types

  int ii = map[i];
  int jj = map[j];

  // use C-C values for these cutoffs since C atoms are biggest

  // cut3rebo = 3 REBO distances

  cut3rebo = 3.0 * rcmax[1][1];

  // cutljrebosq = furthest distance from an owned atom a ghost atom can be
  //               to need its REBO neighs computed
  // interaction = M-K-I-J-L-N with I = owned and J = ghost
  //   this insures N is in the REBO neigh list of L
  //   since I-J < rcLJmax and J-L < rmax

   double cutljrebo = rcLJmax[0][0] + rcmax[1][1];
   cutljrebosq = cutljrebo * cutljrebo;

  // cutmax = furthest distance from an owned atom
  //          at which another atom will feel force, i.e. the ghost cutoff
  // for REBO term in potential:
  //   interaction = M-K-I-J-L-N with I = owned and J = ghost
  //   I to N is max distance = 3 REBO distances
  // for LJ term in potential:
  //   short interaction = M-K-I-J-L-N with I = owned, J = ghost, I-J < rcLJmax
  //   rcLJmax + 2*rcmax, since I-J < rcLJmax and J-L,L-N = REBO distances
  //   long interaction = I-J with I = owned and J = ghost
```

```
  //   cutlj*sigma, since I-J < LJ cutoff

  double cutmax = cut3rebo;
  // if (ljflag) {
 //   cutmax = MAX(cutmax,rcLJmax[0][0] + 2.0*rcmax[0][0]);
 //   cutmax = MAX(cutmax,cutlj*sigma[0][0]);
 // }

  cutsq[i][j] = cutmax * cutmax;
  //cutljsq[ii][jj] = cutlj*sigma[ii][jj] * cutlj*sigma[ii][jj];
  //lj1[ii][jj] = 48.0 * epsilon[ii][jj] * pow(sigma[ii][jj],12.0);
  //lj2[ii][jj] = 24.0 * epsilon[ii][jj] * pow(sigma[ii][jj],6.0);
  //lj3[ii][jj] = 4.0 * epsilon[ii][jj] * pow(sigma[ii][jj],12.0);
  //lj4[ii][jj] = 4.0 * epsilon[ii][jj] * pow(sigma[ii][jj],6.0);

  cutsq[j][i] = cutsq[i][j];
  //cutljsq[jj][ii] = cutljsq[ii][jj];
  //lj1[jj][ii] = lj1[ii][jj];
  //lj2[jj][ii] = lj2[ii][jj];
  //lj3[jj][ii] = lj3[ii][jj];
  //lj4[jj][ii] = lj4[ii][jj];

  return cutmax;
}

/* ----------------------------------------------------------------------
   create REBO neighbor list from main neighbor list
   REBO neighbor list stores neighbors of ghost atoms
------------------------------------------------------------------------- */

void PairBRENNER::REBO_neigh()
{
  int i,j,ii,jj,m,n,inum,jnum,itype,jtype;
  double xtmp,ytmp,ztmp,delx,dely,delz,rsq,dS;
  int *ilist,*jlist,*numneigh,**firstneigh;
  int *neighptr;

  double **x = atom->x;
  int *type = atom->type;
  int nlocal = atom->nlocal;
  int nall = nlocal + atom->nghost;

  if (nall > maxlocal) {
    maxlocal = atom->nmax;
    memory->sfree(REBO_numneigh);
    memory->sfree(REBO_firstneigh);
    memory->sfree(nC);
    memory->sfree(nH);
    REBO_numneigh = (int *)
      memory->smalloc(maxlocal*sizeof(int),"BRENNER:numneigh");
    REBO_firstneigh = (int **)
      memory->smalloc(maxlocal*sizeof(int *),"BRENNER:firstneigh");
    nC = (double *) memory->smalloc(maxlocal*sizeof(double),"BRENNER:nC");
    nH = (double *) memory->smalloc(maxlocal*sizeof(double),"BRENNER:nH");
  }

  inum = list->inum;
  ilist = list->ilist;
  numneigh = list->numneigh;
  firstneigh = list->firstneigh;

  // initialize ghost atom references to -1
```

```
for (i = nlocal; i < nall; i++) REBO_numneigh[i] = -1;

// store all REBO neighs of owned atoms
// scan full neighbor list of I
// if J is ghost and within LJ cutoff:
//   flag it via REBO_numneigh so its REBO neighbors will be stored below
//   REBO requires neighbors of neighbors of i,j in each i,j LJ interaction

npage = 0;
int npnt = 0;

for (ii = 0; ii < inum; ii++) {
  i = ilist[ii];

  if (pgsize - npnt < oneatom) {
    npnt = 0;
    npage++;
    if (npage == maxpage) add_pages(npage);
  }
  neighptr = &pages[npage][npnt];
  n = 0;

  xtmp = x[i][0];
  ytmp = x[i][1];
  ztmp = x[i][2];
  itype = map[type[i]];
  nC[i] = nH[i] = 0.0;
  jlist = firstneigh[i];
  jnum = numneigh[i];

  for (jj = 0; jj < jnum; jj++) {
    j = jlist[jj];
    jtype = map[type[j]];
    delx = xtmp - x[j][0];
    dely = ytmp - x[j][1];
    delz = ztmp - x[j][2];
    rsq = delx*delx + dely*dely + delz*delz;

    if (rsq < rcmaxsq[itype][jtype]) {
     neighptr[n++] = j;
     if (jtype == 0)
       nC[i] += Sp(sqrt(rsq),rcmin[itype][jtype],rcmax[itype][jtype],dS);
     else
       nH[i] += Sp(sqrt(rsq),rcmin[itype][jtype],rcmax[itype][jtype],dS);
    }
    if (j >= nlocal && rsq < cutljrebosq) REBO_numneigh[j] = i;
  }

  REBO_firstneigh[i] = neighptr;
  REBO_numneigh[i] = n;
  npnt += n;
  if (npnt >= pgsize)
    error->one("Neighbor list overflow, boost neigh_modify one or page");
}

// store REBO neighs of ghost atoms that have been flagged in REBO_numneigh
// find by scanning full neighbor list of owned atom M that is neighbor of I

for (i = nlocal; i < nall; i++) {

  if (pgsize - npnt < oneatom) {
    npnt = 0;
    npage++;
```

```
      if (npage == maxpage) add_pages(npage);
    }
    neighptr = &pages[npage][npnt];
    n = 0;

    xtmp = x[i][0];
    ytmp = x[i][1];
    ztmp = x[i][2];
    itype = map[type[i]];
    nC[i] = nH[i] = 0.0;
    m = REBO_numneigh[i];
    if (m < 0) {
      REBO_firstneigh[i] = neighptr;
      REBO_numneigh[i] = 0;
      continue;
    }

    jtype = map[type[m]];
    delx = xtmp - x[m][0];
    dely = ytmp - x[m][1];
    delz = ztmp - x[m][2];
    rsq = delx*delx + dely*dely + delz*delz;

    neighptr[n++] = m;
    if (jtype == 0)
      nC[i] += Sp(sqrt(rsq),rcmin[itype][jtype],rcmax[itype][jtype],dS);
    else
      nH[i] += Sp(sqrt(rsq),rcmin[itype][jtype],rcmax[itype][jtype],dS);

    jlist = firstneigh[m];
    jnum = numneigh[m];

    for (jj = 0; jj < jnum; jj++) {
      j = jlist[jj];
      if (j == i) continue;
      jtype = map[type[j]];
      delx = xtmp - x[j][0];
      dely = ytmp - x[j][1];
      delz = ztmp - x[j][2];
      rsq = delx*delx + dely*dely + delz*delz;

      if (rsq < rcmaxsq[itype][jtype]) {
       neighptr[n++] = j;
       if (jtype == 0)
         nC[i] += Sp(sqrt(rsq),rcmin[itype][jtype],rcmax[itype][jtype],dS);
       else
         nH[i] += Sp(sqrt(rsq),rcmin[itype][jtype],rcmax[itype][jtype],dS);
      }
    }

    REBO_firstneigh[i] = neighptr;
    REBO_numneigh[i] = n;
    npnt += n;
    if (npnt >= pgsize)
      error->one("Neighbor list overflow, boost neigh_modify one or page");
  }
}

/* ----------------------------------------------------------------------
   REBO forces and energy
------------------------------------------------------------------------- */

void PairBRENNER::FREBO(int eflag, int vflag)
```

```
{
  int i,j,k,m,ii,inum,itype,jtype;
  double delx,dely,delz,evdwl,fpair;
  double rsq,rij,wij;
  double Qij,Aij,alphaij,VR,pre,dVRdi,VA,term,bij,dVAdi,dVA;
  double dwij,del[3];
  double Rij,Dij,beta,Sij;
  double Atij,atij,Btij,btij;
  int *ilist,*REBO_neighs;

  evdwl = 0.0;

  double **x = atom->x;
  double **f = atom->f;
  int *type = atom->type;
  int *tag = atom->tag;
  int nlocal = atom->nlocal;
  int newton_pair = force->newton_pair;

  inum = list->inum;
  ilist = list->ilist;

  // two-body interactions from REBO neighbor list, skip half of them

  for (ii = 0; ii < inum; ii++) {
    i = ilist[ii];
    itype = map[type[i]];
    REBO_neighs = REBO_firstneigh[i];

    for (k = 0; k < REBO_numneigh[i]; k++) {
      j = REBO_neighs[k];
      if (tag[i] > tag[j]) continue;
      jtype = map[type[j]];

      delx = x[i][0] - x[j][0];
      dely = x[i][1] - x[j][1];
      delz = x[i][2] - x[j][2];
      rsq = delx*delx + dely*dely + delz*delz;
      rij = sqrt(rsq);
      wij = Sp(rij,rcmin[itype][jtype],rcmax[itype][jtype],dwij);
      if (wij <= TOL) continue;

      Rij = R[itype][jtype];
      Dij = D[itype][jtype];
      beta = bet[itype][jtype];
      Sij = S[itype][jtype];

       VR = wij*(Dij/(Sij-1))*exp(-sqrt(2*Sij)*beta*(rij-Rij));
       dVRdi = -sqrt(2*Sij)*beta*VR;
       dVRdi += VR/wij * dwij;

       VA = -wij*(Dij*Sij/(Sij-1))*exp(-sqrt(2/Sij)*beta*(rij-Rij));
       dVA = -sqrt(2/Sij)*beta*VA;
       dVA += VA/wij * dwij;

      del[0] = delx;
      del[1] = dely;
      del[2] = delz;

       bij = bondorder(i,j,del,rij,VA,f,vflag_atom);
      dVAdi = bij*dVA;
       fpair = -(dVRdi+dVAdi) / rij;
```

```
      f[i][0] += delx*fpair;
      f[i][1] += dely*fpair;
      f[i][2] += delz*fpair;
      f[j][0] -= delx*fpair;
      f[j][1] -= dely*fpair;
      f[j][2] -= delz*fpair;

      if (eflag) evdwl = VR + bij*VA;
      if (evflag) ev_tally(i,j,nlocal,newton_pair,
                           evdwl,0.0,fpair,delx,dely,delz);
    }
  }
}

// ---------------------------------------------------------------------
// S'(t) and S(t) cutoff functions
// ---------------------------------------------------------------------

/* ---------------------------------------------------------------------
   cutoff function Sprime
   return cutoff and dX = derivative
------------------------------------------------------------------------- */

double PairBRENNER::Sp(double Xij, double Xmin, double Xmax, double &dX)
{
  double cutoff;

  double t = (Xij-Xmin) / (Xmax-Xmin);
  if (t <= 0.0) {
    cutoff = 1.0;
    dX = 0.0;
  }
  else if (t >= 1.0) {
    cutoff = 0.0;
    dX = 0.0;
  }
  else {
    cutoff = 0.5 * (1.0+cos(PI*t));
    dX = (-0.5*PI*sin(PI*t)) / (Xmax-Xmin);
  }
  return cutoff;
}

/* ---------------------------------------------------------------------
   Bij function
------------------------------------------------------------------------- */

double PairBRENNER::bondorder(int i, int j, double rij[3],
                          double rijmag, double VA,
                          double **f, int vflag_atom)
{
  int atomi,atomj,k,n,l,atomk,atoml,atomn,atom1,atom2,atom3,atom4;
  int itype,jtype,ktype,ltype,ntype;
  double rik[3],rjl[3],rkn[3],rji[3],rki[3],rlj[3],rknmag,dNki,dwjl,bij;
  double NijC,NijH,NjiC,NjiH,wik,dwik,dwkn,wjl;
  double rikmag,rjlmag,cosjik,cosijl,g,tmp2,tmp3;
  double Etmp,pij,tmp,wij,dwij,NconjtmpI,NconjtmpJ,Nki,Nlj,dS;
  double lamdajik,lamdaijl,dgdc,dgdN,pji,Nijconj,piRC;
  double dcosjikdri[3],dcosijldri[3],dcosjikdrk[3];
  double dN2[2],dN3[3];
  double dcosjikdrj[3],dcosijldrj[3],dcosijldrl[3];
  double Tij;
  double r32[3],r32mag,cos321,r43[3],r13[3];
```

```
double dNlj;
double om1234,rln[3];
double rlnmag,dwln,r23[3],r23mag,r21[3],r21mag;
double w21,dw21,r34[3],r34mag,cos234,w34,dw34;
double cross321[3],cross234[3],prefactor,SpN;
double fcijpc,fcikpc,fcjlpc,fcjkpc,fcilpc;
double dt2dik[3],dt2djl[3],dt2dij[3],aa,aaa1,aaa2,at2,cw,cwnum,cwnom;
double sin321,sin234,rr,rijrik,rijrjl,rjk2,rik2,ril2,rjl2;
double dctik,dctjk,dctjl,dctij,dctji,dctil,rik2i,rjl2i,sink2i,sinl2i;
double rjk[3],ril[3],dt1dik,dt1djk,dt1djl,dt1dil,dt1dij;
double F23[3],F12[3],F34[3],F31[3],F24[3],fi[3],fj[3],fk[3],fl[3];
double f1[3],f2[3],f3[3],f4[4];
double dcut321,PijS,PjiS;
double rij2,tspjik,dtsjik,tspijl,dtsijl,costmp;
int *REBO_neighs,*REBO_neighs_i,*REBO_neighs_j,*REBO_neighs_k,*REBO_neighs_l;

double **x = atom->x;
int *type = atom->type;

atomi = i;
atomj = j;
itype = map[type[i]];
jtype = map[type[j]];
wij = Sp(rijmag,rcmin[itype][jtype],rcmax[itype][jtype],dwij);
NijC = nC[i]-(wij*kronecker(jtype,0));
NijH = nH[i]-(wij*kronecker(jtype,1));
NjiC = nC[j]-(wij*kronecker(itype,0));
NjiH = nH[j]-(wij*kronecker(itype,1));
bij = 0.0;
tmp = 0.0;
tmp2 = 0.0;
tmp3 = 0.0;
dgdc = 0.0;
dgdN = 0.0;
NconjtmpI = 0.0;
NconjtmpJ = 0.0;
Etmp = 0.0;

double delta;
delta = del[itype][itype];

REBO_neighs = REBO_firstneigh[i];
for (k = 0; k < REBO_numneigh[i]; k++) {
  atomk = REBO_neighs[k];
  if (atomk != atomj) {
    ktype = map[type[atomk]];
    rik[0] = x[atomi][0]-x[atomk][0];
    rik[1] = x[atomi][1]-x[atomk][1];
    rik[2] = x[atomi][2]-x[atomk][2];
    rikmag = sqrt((rik[0]*rik[0])+(rik[1]*rik[1])+(rik[2]*rik[2]));
    lamdajik = 4.0*kronecker(itype,1) *
     ((rho[ktype][1]-rikmag)-(rho[jtype][1]-rijmag));
    wik = Sp(rikmag,rcmin[itype][ktype],rcmax[itype][ktype],dS);
    Nki = nC[atomk]-(wik*kronecker(itype,0))+nH[atomk] -
     (wik*kronecker(itype,1));
    cosjik = ((rij[0]*rik[0])+(rij[1]*rik[1])+(rij[2]*rik[2])) /
     (rijmag*rikmag);
    cosjik = MIN(cosjik,1.0);
    cosjik = MAX(cosjik,-1.0);

    // evaluate splines g and derivatives dg

    g = gSpline(cosjik,(NijC+NijH),itype,&dgdc,&dgdN);
```

```
    Etmp = Etmp+(wik*g);
    tmp3 = tmp3+(wik*dgdN*exp(lamdajik));
  }
}

PijS = 0.0;
dN2[0] = 0.0;
dN2[1] = 0.0;

pij = pow(1.0+Etmp,-delta);
tmp = -delta*pow(1.0+Etmp,-delta-1.0);

// pij forces

REBO_neighs = REBO_firstneigh[i];
for (k = 0; k < REBO_numneigh[i]; k++) {
  atomk = REBO_neighs[k];
  if (atomk != atomj) {
    ktype = map[type[atomk]];
    rik[0] = x[atomi][0]-x[atomk][0];
    rik[1] = x[atomi][1]-x[atomk][1];
    rik[2] = x[atomi][2]-x[atomk][2];
    rikmag = sqrt((rik[0]*rik[0])+(rik[1]*rik[1])+(rik[2]*rik[2]));
    lamdajik = 4.0*kronecker(itype,1) *
     ((rho[ktype][1]-rikmag)-(rho[jtype][1]-rijmag));
    wik = Sp(rikmag,rcmin[itype][ktype],rcmax[itype][ktype],dwik);
    cosjik = (rij[0]*rik[0] + rij[1]*rik[1] + rij[2]*rik[2]) /
     (rijmag*rikmag);
    cosjik = MIN(cosjik,1.0);
    cosjik = MAX(cosjik,-1.0);

    dcosjikdri[0] = ((rij[0]+rik[0])/(rijmag*rikmag)) -
     (cosjik*((rij[0]/(rijmag*rijmag))+(rik[0]/(rikmag*rikmag))));
    dcosjikdri[1] = ((rij[1]+rik[1])/(rijmag*rikmag)) -
     (cosjik*((rij[1]/(rijmag*rijmag))+(rik[1]/(rikmag*rikmag))));
    dcosjikdri[2] = ((rij[2]+rik[2])/(rijmag*rikmag)) -
     (cosjik*((rij[2]/(rijmag*rijmag))+(rik[2]/(rikmag*rikmag))));
    dcosjikdrk[0] = (-rij[0]/(rijmag*rikmag)) +
     (cosjik*(rik[0]/(rikmag*rikmag)));
    dcosjikdrk[1] = (-rij[1]/(rijmag*rikmag)) +
     (cosjik*(rik[1]/(rikmag*rikmag)));
    dcosjikdrk[2] = (-rij[2]/(rijmag*rikmag)) +
     (cosjik*(rik[2]/(rikmag*rikmag)));
    dcosjikdrj[0] = (-rik[0]/(rijmag*rikmag)) +
     (cosjik*(rij[0]/(rijmag*rijmag)));
    dcosjikdrj[1] = (-rik[1]/(rijmag*rikmag)) +
     (cosjik*(rij[1]/(rijmag*rijmag)));
    dcosjikdrj[2] = (-rik[2]/(rijmag*rikmag)) +
     (cosjik*(rij[2]/(rijmag*rijmag)));

    g = gSpline(cosjik,(NijC+NijH),itype,&dgdc,&dgdN);
    tmp2 = VA*.5*(tmp*wik*dgdc);
    fj[0] = -tmp2*dcosjikdrj[0];
    fj[1] = -tmp2*dcosjikdrj[1];
    fj[2] = -tmp2*dcosjikdrj[2];
    fi[0] = -tmp2*dcosjikdri[0];
    fi[1] = -tmp2*dcosjikdri[1];
    fi[2] = -tmp2*dcosjikdri[2];
    fk[0] = -tmp2*dcosjikdrk[0];
    fk[1] = -tmp2*dcosjikdrk[1];
    fk[2] = -tmp2*dcosjikdrk[2];

    // coordination forces
```

```
      // dwik forces

      tmp2 = VA*.5*(tmp*dwik*g)/rikmag;
      fi[0] -= tmp2*rik[0];
      fi[1] -= tmp2*rik[1];
      fi[2] -= tmp2*rik[2];
      fk[0] += tmp2*rik[0];
      fk[1] += tmp2*rik[1];
      fk[2] += tmp2*rik[2];

      f[atomi][0] += fi[0]; f[atomi][1] += fi[1]; f[atomi][2] += fi[2];
      f[atomj][0] += fj[0]; f[atomj][1] += fj[1]; f[atomj][2] += fj[2];
      f[atomk][0] += fk[0]; f[atomk][1] += fk[1]; f[atomk][2] += fk[2];

      if (vflag_atom) {
       rji[0] = -rij[0]; rji[1] = -rij[1]; rji[2] = -rij[2];
       rki[0] = -rik[0]; rki[1] = -rik[1]; rki[2] = -rik[2];
       v_tally3(atomi,atomj,atomk,fj,fk,rji,rki);
      }
    }
  }

  tmp = 0.0;
  tmp2 = 0.0;
  tmp3 = 0.0;
  Etmp = 0.0;

 delta = del[jtype][jtype];

  REBO_neighs = REBO_firstneigh[j];
  for (l = 0; l < REBO_numneigh[j]; l++) {
    atoml = REBO_neighs[l];
    if (atoml != atomi) {
      ltype = map[type[atoml]];
      rjl[0] = x[atomj][0]-x[atoml][0];
      rjl[1] = x[atomj][1]-x[atoml][1];
      rjl[2] = x[atomj][2]-x[atoml][2];
      rjlmag = sqrt((rjl[0]*rjl[0])+(rjl[1]*rjl[1])+(rjl[2]*rjl[2]));
      lamdaijl = 4.0*kronecker(jtype,1) *
       ((rho[ltype][1]-rjlmag)-(rho[itype][1]-rijmag));
      wjl = Sp(rjlmag,rcmin[jtype][ltype],rcmax[jtype][ltype],dS);
      Nlj = nC[atoml]-(wjl*kronecker(jtype,0)) +
       nH[atoml]-(wjl*kronecker(jtype,1));
      cosijl = -1.0*((rij[0]*rjl[0])+(rij[1]*rjl[1])+(rij[2]*rjl[2])) /
       (rijmag*rjlmag);
      cosijl = MIN(cosijl,1.0);
      cosijl = MAX(cosijl,-1.0);

      // evaluate splines g and derivatives dg

      g = gSpline(cosijl,NjiC+NjiH,jtype,&dgdc,&dgdN);
      Etmp = Etmp+(wjl*g);
      tmp3 = tmp3+(wjl*dgdN*exp(lamdaijl));
    }
  }

  PjiS = 0.0;
  dN2[0] = 0.0;
  dN2[1] = 0.0;

  pji = pow(1.0+Etmp,-delta);
  tmp = -delta*pow(1.0+Etmp,-delta-1.0);
```

```
REBO_neighs = REBO_firstneigh[j];
for (l = 0; l < REBO_numneigh[j]; l++) {
  atoml = REBO_neighs[l];
  if (atoml != atomi) {
    ltype = map[type[atoml]];
    rjl[0] = x[atomj][0]-x[atoml][0];
    rjl[1] = x[atomj][1]-x[atoml][1];
    rjl[2] = x[atomj][2]-x[atoml][2];
    rjlmag = sqrt((rjl[0]*rjl[0])+(rjl[1]*rjl[1])+(rjl[2]*rjl[2]));
    lamdaijl = 4.0*kronecker(jtype,1) *
     ((rho[ltype][1]-rjlmag)-(rho[itype][1]-rijmag));
    wjl = Sp(rjlmag,rcmin[jtype][ltype],rcmax[jtype][ltype],dwjl);
    cosijl = (-1.0*((rij[0]*rjl[0])+(rij[1]*rjl[1])+(rij[2]*rjl[2]))) /
     (rijmag*rjlmag);
    cosijl = MIN(cosijl,1.0);
    cosijl = MAX(cosijl,-1.0);

    dcosijldri[0] = (-rjl[0]/(rijmag*rjlmag)) -
     (cosijl*rij[0]/(rijmag*rijmag));
    dcosijldri[1] = (-rjl[1]/(rijmag*rjlmag)) -
     (cosijl*rij[1]/(rijmag*rijmag));
    dcosijldri[2] = (-rjl[2]/(rijmag*rjlmag)) -
     (cosijl*rij[2]/(rijmag*rijmag));
    dcosijldrj[0] = ((-rij[0]+rjl[0])/(rijmag*rjlmag)) +
     (cosijl*((rij[0]/pow(rijmag,2.0))-(rjl[0]/(rjlmag*rjlmag))));
    dcosijldrj[1] = ((-rij[1]+rjl[1])/(rijmag*rjlmag)) +
     (cosijl*((rij[1]/pow(rijmag,2.0))-(rjl[1]/(rjlmag*rjlmag))));
    dcosijldrj[2] = ((-rij[2]+rjl[2])/(rijmag*rjlmag)) +
     (cosijl*((rij[2]/pow(rijmag,2.0))-(rjl[2]/(rjlmag*rjlmag))));
    dcosijldrl[0] = (rij[0]/(rijmag*rjlmag))+(cosijl*rjl[0]/(rjlmag*rjlmag));
    dcosijldrl[1] = (rij[1]/(rijmag*rjlmag))+(cosijl*rjl[1]/(rjlmag*rjlmag));
    dcosijldrl[2] = (rij[2]/(rijmag*rjlmag))+(cosijl*rjl[2]/(rjlmag*rjlmag));

    // evaluate splines g and derivatives dg

    g = gSpline(cosijl,NjiC+NjiH,jtype,&dgdc,&dgdN);
    tmp2 = VA*.5*(tmp*wjl*dgdc);
    fi[0] = -tmp2*dcosijldri[0];
    fi[1] = -tmp2*dcosijldri[1];
    fi[2] = -tmp2*dcosijldri[2];
    fj[0] = -tmp2*dcosijldrj[0];
    fj[1] = -tmp2*dcosijldrj[1];
    fj[2] = -tmp2*dcosijldrj[2];
    fl[0] = -tmp2*dcosijldrl[0];
    fl[1] = -tmp2*dcosijldrl[1];
    fl[2] = -tmp2*dcosijldrl[2];

    // coordination forces

    // dwik forces

    tmp2 = VA*.5*(tmp*dwjl*g)/rjlmag;
    fj[0] -= tmp2*rjl[0];
    fj[1] -= tmp2*rjl[1];
    fj[2] -= tmp2*rjl[2];
    fl[0] += tmp2*rjl[0];
    fl[1] += tmp2*rjl[1];
    fl[2] += tmp2*rjl[2];

    f[atomi][0] += fi[0]; f[atomi][1] += fi[1]; f[atomi][2] += fi[2];
    f[atomj][0] += fj[0]; f[atomj][1] += fj[1]; f[atomj][2] += fj[2];
    f[atoml][0] += fl[0]; f[atoml][1] += fl[1]; f[atoml][2] += fl[2];
```

```
      if (vflag_atom) {
       rlj[0] = -rjl[0]; rlj[1] = -rjl[1]; rlj[2] = -rjl[2];
       v_tally3(atomi,atomj,atoml,fi,fl,rij,rlj);
      }
    }
  }

  bij = (0.5*(pij+pji));
  return bij;
}

/* ----------------------------------------------------------------------
   G spline
------------------------------------------------------------------------ */

double PairBRENNER::gSpline(double costh, double Nij, int typei,
                        double *dgdc, double *dgdN)
{
  double coeffs[6],dS,g1,g2,dg1,dg2,cut,g;
  int i,j;

  i = 0;
  j = 0;
  g = 0.0;
  cut = 0.0;
  dS = 0.0;
  dg1 = 0.0;
  dg2 = 0.0;
  *dgdc = 0.0;
  *dgdN = 0.0;

  double a0,c02,d02,denom,h;

  a0 = a[typei][typei];
  c02 = c[typei][typei];
  d02 = d[typei][typei];

  denom = d02+pow((1+costh),2);
  g = a0*(1+c02/d02-c02/denom);
  *dgdc = a0*c02*(pow(denom,-2))*2*(1+costh);
  *dgdN = 0;

  return g;
}

/* ----------------------------------------------------------------------
   Kronecker delta function
------------------------------------------------------------------------ */

double PairBRENNER::kronecker(int a, int b)
{
  double kd;
  if (a == b) kd = 1.0;
  else kd = 0.0;
  return kd;
}

/* ----------------------------------------------------------------------
   add pages to REBO neighbor list, starting at npage
------------------------------------------------------------------------ */

void PairBRENNER::add_pages(int npage)
```

```
{
  maxpage += PGDELTA;
  pages = (int **)
    memory->srealloc(pages,maxpage*sizeof(int *),"BRENNER:pages");
  for (int i = npage; i < maxpage; i++)
    pages[i] = (int *) memory->smalloc(pgsize*sizeof(int),"BRENNER:pages[i]");
}

/* ----------------------------------------------------------------------
   read BRENNER potential file
------------------------------------------------------------------------- */

void PairBRENNER::read_file(char *filename)
{
  int i,j,k,l,limit;
  char s[MAXLINE];

  // REBO Parameters (BRENNER)

  double rcmin_CC,rcmin_CH,rcmin_HH,rcmax_CC,rcmax_CH,
    rcmax_HH,rcmaxp_CC,rcmaxp_CH,rcmaxp_HH;
  double Q_CC,Q_CH,Q_HH,alpha_CC,alpha_CH,alpha_HH,A_CC,A_CH,A_HH;
  double BIJc_CC1,BIJc_CC2,BIJc_CC3,BIJc_CH1,BIJc_CH2,BIJc_CH3,
    BIJc_HH1,BIJc_HH2,BIJc_HH3;
  double Beta_CC1,Beta_CC2,Beta_CC3,Beta_CH1,Beta_CH2,Beta_CH3,
    Beta_HH1,Beta_HH2,Beta_HH3;
  double rho_CC,rho_CH,rho_HH;

  // Brenner Parameters

  double R_CC,D_CC,beta_CC,S_CC,delta_CC,alp_CC,rmin_CC,rmax_CC;
  double a0_CC,c02_CC,d02_CC;
  double R_SS,D_SS,beta_SS,S_SS,delta_SS,alp_SS,rmin_SS,rmax_SS;
  double a0_SS,c02_SS,d02_SS;
  double R_CS,D_CS,beta_CS,S_CS,delta_CS,alp_CS,rmin_CS,rmax_CS;
  double a0_CS,c02_CS,d02_CS;

  // LJ Parameters (BRENNER)

  double rcLJmin_CC,rcLJmin_CH,rcLJmin_HH,rcLJmax_CC,rcLJmax_CH,
    rcLJmax_HH,bLJmin_CC;
  double bLJmin_CH,bLJmin_HH,bLJmax_CC,bLJmax_CH,bLJmax_HH,
    epsilon_CC,epsilon_CH,epsilon_HH;
  double sigma_CC,sigma_CH,sigma_HH,epsilonT_CCCC,epsilonT_CCCH,epsilonT_HCCH;

  MPI_Comm_rank(world,&me);

  // read file on proc 0

  if (me == 0) {
    FILE *fp = fopen(filename,"r");
    if (fp == NULL) {
      char str[128];
      sprintf(str,"Cannot open BRENNER potential file %s",filename);
      error->one(str);
    }

    // skip initial comment lines

    while (1) {
      fgets(s,MAXLINE,fp);
      if (s[0] != '#') break;
    }
```

```
// read parameters

fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&R_CC);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&D_CC);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&beta_CC);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&S_CC);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&delta_CC);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&alp_CC);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&rmin_CC);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&rmax_CC);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&a0_CC);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&c02_CC);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&d02_CC);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&R_SS);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&D_SS);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&beta_SS);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&S_SS);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&delta_SS);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&alp_SS);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&rmin_SS);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&rmax_SS);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&a0_SS);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&c02_SS);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&d02_SS);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&R_CS);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&D_CS);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&beta_CS);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&S_CS);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&delta_CS);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&alp_CS);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&rmin_CS);
fgets(s,MAXLINE,fp);
sscanf(s,"%lg",&rmax_CS);
```

```
  fgets(s,MAXLINE,fp);
  sscanf(s,"%lg",&a0_CS);
  fgets(s,MAXLINE,fp);
  sscanf(s,"%lg",&c02_CS);
  fgets(s,MAXLINE,fp);
  sscanf(s,"%lg",&d02_CS);
}

// store read-in values in arrays

if (me == 0) {

  // REBO

  R[0][0] = R_CC;
  D[0][0] = D_CC;
  bet[0][0] = beta_CC;
  S[0][0] = S_CC;
  del[0][0] = delta_CC;
  alp[0][0] = alp_CC;
  rmin[0][0] = rmin_CC;
  rmax[0][0] = rmax_CC;
  a[0][0] = a0_CC;
  c[0][0] = c02_CC;
  d[0][0] = d02_CC;

  R[1][1] = R_SS;
  D[1][1] = D_SS;
  bet[1][1] = beta_SS;
  S[1][1] = S_SS;
  del[1][1] = delta_SS;
  alp[1][1] = alp_SS;
  rmin[1][1] = rmin_SS;
  rmax[1][1] = rmax_SS;
  a[1][1] = a0_SS;
  c[1][1] = c02_SS;
  d[1][1] = d02_SS;

  R[0][1] = R_CS;
  D[0][1] = D_CS;
  bet[0][1] = beta_CS;
  S[0][1] = S_CS;
  del[0][1] = delta_CS;
  alp[0][1] = alp_CS;
  rmin[0][1] = rmin_CS;
  rmax[0][1] = rmax_CS;
  a[0][1] = a0_CS;
  c[0][1] = c02_CS;
  d[0][1] = d02_CS;

  R[1][0] = R[0][1];
  D[1][0] = D[0][1];
  bet[1][0] = bet[0][1];
  S[1][0] = S[0][1];
  del[1][0] = del[0][1];
  alp[1][0] = alp[0][1];
  rmin[1][0] = rmin[0][1];
  rmax[1][0] = rmax[0][1];
  a[1][0] = a[0][1];
  c[1][0] = c[0][1];
  d[1][0] = d[0][1];

  rcmin[0][0] = rmin_CC;
```

```
    rcmin[0][1] = rmin_CS;
    rcmin[1][0] = rmin_CS;
    rcmin[1][1] = rmin_SS;

    rcmax[0][0] = rmax_CC;
    rcmax[0][1] = rmax_CS;
    rcmax[1][0] = rmax_CS;
    rcmax[1][1] = rmax_SS;

    rcmaxsq[0][0] = rcmax[0][0]*rcmax[0][0];
    rcmaxsq[1][0] = rcmax[1][0]*rcmax[1][0];
    rcmaxsq[0][1] = rcmax[0][1]*rcmax[0][1];
    rcmaxsq[1][1] = rcmax[1][1]*rcmax[1][1];

    rcLJmax[0][0] = 3.816370964;
    rcLJmax[0][1] = 2.974524428;
    rcLJmax[1][0] = 3.395447696;
    rcLJmax[1][1] = 3.395447696;

    rcLJmaxsq[0][0] = rcLJmax[0][0]*rcLJmax[0][0];
    rcLJmaxsq[1][0] = rcLJmax[1][0]*rcLJmax[1][0];
    rcLJmaxsq[0][1] = rcLJmax[0][1]*rcLJmax[0][1];
    rcLJmaxsq[1][1] = rcLJmax[1][1]*rcLJmax[1][1];
  }

  // broadcast read-in and setup values

  MPI_Bcast(&rcmin[0][0],4,MPI_DOUBLE,0,world);
  MPI_Bcast(&rcmax[0][0],4,MPI_DOUBLE,0,world);
  MPI_Bcast(&R[0][0],4,MPI_DOUBLE,0,world);
  MPI_Bcast(&D[0][0],4,MPI_DOUBLE,0,world);
  MPI_Bcast(&bet[0][0],4,MPI_DOUBLE,0,world);
  MPI_Bcast(&S[0][0],4,MPI_DOUBLE,0,world);
  MPI_Bcast(&del[0][0],4,MPI_DOUBLE,0,world);
  MPI_Bcast(&alp[0][0],4,MPI_DOUBLE,0,world);
  MPI_Bcast(&rmin[0][0],4,MPI_DOUBLE,0,world);
  MPI_Bcast(&rmax[0][0],4,MPI_DOUBLE,0,world);
  MPI_Bcast(&a[0][0],4,MPI_DOUBLE,0,world);
  MPI_Bcast(&c[0][0],4,MPI_DOUBLE,0,world);
  MPI_Bcast(&d[0][0],4,MPI_DOUBLE,0,world);
  MPI_Bcast(&rcmaxsq[0][0],4,MPI_DOUBLE,0,world);
  MPI_Bcast(&rcLJmax[0][0],4,MPI_DOUBLE,0,world);
  MPI_Bcast(&rcLJmaxsq[0][0],4,MPI_DOUBLE,0,world);
}

/* ----------------------------------------------------------------------
   memory usage of local atom-based arrays
------------------------------------------------------------------------- */

double PairBRENNER::memory_usage()
{
  double bytes = 0.0;
  bytes += maxlocal * sizeof(int);
  bytes += maxlocal * sizeof(int *);
  bytes += maxpage * neighbor->pgsize * sizeof(int);
  bytes += 2 * maxlocal * sizeof(double);
  return bytes;
}
```

REFERENCES

[1] Rockwell Collins, Inc., 2009.

[2] J. Liu, B. Michel, M. Rencz, C. Tantolin, C. Sarno, R. Miessner, K.-V. Schuett, X. Tang, S. Demoustier, and A. Ziaei, "Recent progress of thermal interface material research - an overview," Rome, Italy, 2008, pp. 156-162.

[3] F. Sarvar, D. C. Whalley, and P. P. Conway, "Thermal interface materials - A review of the state of the art," Dresden, Saxony, Germany, 2007, pp. 1292-1302.

[4] R. Prasher, "Thermal interface materials: historical perspective, status, and future directions," *Proceedings of the IEEE,* vol. 94, pp. 1571-86, 2006.

[5] C. I. Chen, C. Y. Ni, C. M. Chang, D. S. Liu, H. Y. Pan, and T. D. Yuan, "Thermal characterization of thermal interface materials," *Experimental Techniques,* vol. 32, pp. 48-52, 2008.

[6] ComputerShopper.com, "Build Your Own Budget Gaming PC: Putting It All Together," 2009.

[7] S. R. Mirmira, E. E. Marotta, and L. S. Fletcher, "Thermal contact conductance of adhesives for microelectronic systems," *Journal of thermophysics and heat transfer,* vol. 11, pp. 141-145, 1997.

[8] D. D. L. Chung, "Thermal interface materials," *Journal of Materials Engineering and Performance,* vol. 10, pp. 56-59, 2001.

[9] C.-P. Chiu, J. G. Maveety, and Q. A. Tran, "Characterization of solder interfaces using laser flash metrology," *Microelectronics Reliability,* vol. 42, pp. 93-100, 2002.

[10] M. J. Biercuk, M. C. Llaguno, M. Radosavljevic, J. K. Hyun, A. T. Johnson, and J. E. Fischer, "Carbon nanotube composites for thermal management," *Applied Physics Letters,* vol. 80, pp. 2767-9, 2002.

[11] C.-W. Nan, G. Liu, Y. Lin, and M. Li, "Interface effect on thermal conductivity of carbon nanotube composites," *Applied Physics Letters,* vol. 85, pp. 3549-3551, 2004.

[12] X. Hu, L. Jiang, and K. E. Goodson, "Thermal conductance enhancement of particle-filled thermal interface materials using carbon nanotube inclusions," Las Vegas, NV, United states, 2004, pp. 63-69.

[13] C. H. Liu, H. Huang, Y. Wu, and S. S. Fan, "Thermal conductivity improvement of silicone elastomer with carbon nanotube loading," *Applied Physics Letters,* vol. 84, pp. 4248-4250, 2004.

[14] K. M. Razeeb, A. Munari, E. Dalton, J. Punch, and S. Roy, "Thermal properties of carbon nanotube-polymer composites for thermal interface material applications," Vancouver, BC, Canada, 2007, pp. 817-823.

[15] L. Zhu, Y. Sun, J. Xu, Z. Zhang, D. W. Hess, and C. P. Wong, "Aligned carbon nanotubes for electrical interconnect and thermal management," Lake Buena Vista, FL, United States, 2005, pp. 44-50.

[16] X. J. Hu, A. A. Padilla, J. Xu, T. S. Fisher, and K. E. Goodson, "3-omega measurements of vertically oriented carbon nanotubes on silicon," *Journal of Heat Transfer,* vol. 128, pp. 1109-1113, 2006.

[17] X. Hu, A. A. Padilla, J. Xu, T. S. Fisher, and K. E. Goodson, "Thermal characterization of vertically-oriented carbon nanotubes on silicon," San Jose, CA, United States, 2005, pp. 292-297.
[18] X. J. Hu, M. A. Panzer, and K. E. Goodson, "Infrared microscopy thermal characterization of opposing carbon nanotube arrays," *Journal of Heat Transfer,* vol. 129, pp. 91-93, 2007.
[19] M. A. Panzer, G. Zhang, D. Mann, X. Hu, E. Pop, H. Dai, and K. E. Goodson, "Thermal Properties of Metal-Coated Vertically Aligned Single-Wall Nanotube Arrays," *Journal of Heat Transfer,* vol. 130, pp. 052401-9, 2008.
[20] J. Xu and T. S. Fisher, "Enhanced thermal contact conductance using carbon nanotube arrays," Las Vegas, NV, USA, 2004, pp. 549-55.
[21] J. Xu and T. S. Fisher, "Enhancement of thermal interface materials with carbon nanotube arrays," *International Journal of Heat and Mass Transfer,* vol. 49, pp. 1658-1666, 2006.
[22] X. Jun and T. S. Fisher, "Enhanced thermal contact conductance using carbon nanotube array interfaces," *Components and Packaging Technologies, IEEE Transactions on,* vol. 29, pp. 261-267, 2006.
[23] B. A. Cola, J. Xu, C. Cheng, X. Xu, T. S. Fisher, and H. Hu, "Photoacoustic characterization of carbon nanotube array thermal interfaces," *Journal of Applied Physics,* vol. 101, p. 054313, 2007.
[24] P. B. Amama, B. A. Cola, T. D. Sands, X. Xu, and T. S. Fisher, "Dendrimer-assisted controlled growth of carbon nanotubes for enhanced thermal interface conductance," *Nanotechnology,* vol. 18, 2007.
[25] B. A. Cola, X. Xianfan, and T. S. Fisher, "Increased real contact in thermal interfaces: a carbon nanotube/foil material," *Applied Physics Letters,* vol. 90, pp. 93513-1, 2007.
[26] B. A. Cola, P. B. Amama, X. Xianfan, and T. S. Fisher, "Effects of growth temperature on carbon nanotube array thermal interfaces," *Journal of Heat Transfer,* vol. 130, pp. 114503-1, 2008.
[27] B. A. Cola, X. F. Xu, T. S. Fisher, M. A. Capano, and P. B. Amama, "Carbon nanotube array thermal interfaces for high-temperature silicon carbide devices," *Nanoscale and Microscale Thermophysical Engineering,* vol. 12, pp. 228-237, 2008.
[28] T. Tong, Y. Zhao, L. Delzeit, A. Kashani, M. Meyyappan, and A. Majumdar, "Dense vertically aligned multiwalled carbon nanotube arrays as thermal interface materials," *IEEE Transactions on Components and Packaging Technologies,* vol. 30, pp. 92-100, 2007.
[29] Y. Son, S. K. Pal, T. Borca-Tasciuc, P. M. Ajayan, and R. W. Siegel, "Thermal resistance of the native interface between vertically aligned multiwalled carbon nanotube arrays and their SiO2/Si substrate," *Journal of Applied Physics,* vol. 103, Jan 2008.
[30] R. S. Prasher and P. E. Phelan, "Microscopic and macroscopic thermal contact resistances of pressed mechanical contacts," *Journal of Applied Physics,* vol. 100, pp. 063538-8, 2006.
[31] R. Prasher, "Predicting the Thermal Resistance of Nanosized Constrictions," *Nano Letters,* vol. 5, pp. 2155-2159, 2005.

[32] Q. Ngo, B. A. Cruden, A. M. Cassell, G. Sims, M. Meyyappan, J. Li, and C. Y. Yang, "Thermal interface properties of Cu-filled vertically aligned carbon nanofiber arrays," *Nano Letters,* vol. 4, pp. 2403-2407, 2004.
[33] C. Yang, Z. Kai, Z. Min, P. C. H. Chan, and M. M. F. Yuen, "Carbon nanotube/copper composites for via filling and thermal management," Reno, NV, USA, 2007, pp. 1224-9.
[34] Y. Wu, C. H. Liu, H. Huang, and S. S. Fan, "Effects of surface metal layer on the thermal contact resistance of carbon nanotube arrays," *Applied Physics Letters,* vol. 87, pp. 213108-1, 2005.
[35] S. Shaikh, K. Lafdi, and E. Silverman, "The effect of a CNT interface on the thermal resistance of contacting surfaces," *Carbon,* vol. 45, pp. 695-703, 2007.
[36] S. Shaikh, L. Li, K. Lafdi, and J. Huie, "Thermal conductivity of an aligned carbon nanotube array," *Carbon,* vol. 45, pp. 2608-2613, 2007.
[37] L. Zhu, Y. Xiu, D. W. Hess, and C.-P. Wong, "Aligned carbon nanotube stacks by water-assisted selective etching," *Nano Letters,* vol. 5, pp. 2641-2645, 2005.
[38] L. Zhu, K.-S. Moon, B. Bertram, D. W. Hess, and C. P. Wong, "Assembling carbon nanotube bundles using transfer process for fine-pitch electrical interconnect applications," Sparks, NV, United States, 2007, pp. 1981-1985.
[39] L. Zhu, J. Xu, Y. Xiu, Y. Sun, D. W. Hess, and C. P. Wong, "Growth and electrical characterization of high-aspect-ratio carbon nanotube arrays," *Carbon,* vol. 44, pp. 253-258, 2006.
[40] L. Zhu, Y. Xiu, D. W. Hess, and C. P. Wong, "Growth of aligned carbon nanotube arrays for electrical interconnect," Singapore, Singapore, 2005, pp. 646-651.
[41] L. Zhu, Y. Sun, D. W. Hess, and C.-P. Wong, "Well-aligned open-ended carbon nanotube architectures: An approach for device assembly," *Nano Letters,* vol. 6, pp. 243-247, 2006.
[42] L. Zhu, D. W. Hess, and C. P. Wong, "Assembling carbon nanotube films as thermal interface materials," Sparks, NV, United states, 2007, pp. 2006-2010.
[43] W. Lin, Y. Xiu, L. Zhu, K.-S. Moon, and C. P. Wong, "Assembling of carbon nanotube structures by chemical anchoring for packaging applications," Lake Buena Vista, FL, United states, 2008, pp. 421-426.
[44] W. Lin, Y. Xiu, H. Jiang, R. Zhang, O. Hildreth, K.-S. Moon, and C. P. Wong, "Self-Assembled Monolayer-Assisted Chemical Transfer of In Situ Functionalized Carbon Nanotubes," *Journal of the American Chemical Society,* vol. 130, pp. 9636-9637, 2008.
[45] P. Kim, L. Shi, A. Majumdar, and P. L. McEuen, "Thermal transport measurements of individual multiwalled nanotubes," *Physical Review Letters,* vol. 87, pp. 215502-1, 2001.
[46] A. K. Ghatak and Z. S. Kothari, *Introduction to Lattice Dynamics*. London, UK: Addison-Wesley, 1972.
[47] N. Ashcroft and D. Mermin, *Solid State Physics*: Thomson Learning, 1976.
[48] C. Hepburn, Essex, UK, 2009.
[49] M. Dove, *Introduction to Lattice Dynamics*. Great Britain: Cambridge University Press, 1993.

[50] D. C. Rapaport, *The Art of Molecular Dynamics Simulation*: Cambridge University Press, 1996.
[51] S. Volz, J. B. Saulnier, G. Chen, and P. Beauchamp, "Computation of thermal conductivity of Si/Ge superlattices by molecular dynamics techniques," UK, 2000, pp. 815-19.
[52] P. Heino, "Dispersion and thermal resistivity in silicon nanofilms by molecular dynamics," *European Physical Journal B,* vol. 60, pp. 171-179, 2007.
[53] L. Sun and J. Y. Murthy, "Domain size effects in molecular dynamics simulation of phonon transport in silicon," *Applied Physics Letters,* vol. 89, p. 171919, 2006.
[54] Z. Hong and J. B. Freund, "Full-spectrum phonon relaxation times in crystalline Si from molecular dynamics simulations," *Journal of Applied Physics,* vol. 104, pp. 033514-1, 2008.
[55] C. J. Gomes, M. Madrid, J. V. Goicochea, and C. H. Amon, "In-plane and out-of-plane thermal conductivity of silicon thin films predicted by molecular dynamics," *Journal of Heat Transfer,* vol. 128, pp. 1114-1121, 2006.
[56] M. Hu, P. Keblinski, and P. K. Schelling, "Kapitza conductance of silicon--amorphous polyethylene interfaces by molecular dynamics simulations," *Physical Review B (Condensed Matter and Materials Physics),* vol. 79, pp. 104305-7, 2009.
[57] Q. Tang and Y. Yao, "The Kapitza resistance across grain boundary by molecular dynamics simulation," *Nanoscale and Microscale Thermophysical Engineering,* vol. 10, pp. 387-398, 2006.
[58] M. Hu, S. Shenogin, and P. Keblinski, "Molecular dynamics simulation of interfacial thermal conductance between silicon and amorphous polyethylene," *Applied Physics Letters,* vol. 91, p. 241910, 2007.
[59] X.-L. Feng, Z.-X. Li, and Z.-Y. Guo, "Molecular dynamics simulation of thermal conductivity of nanoscale thin silicon films," *Microscale Thermophysical Engineering,* vol. 7, pp. 153-161, 2003.
[60] S. K. Saha and L. Shi, "Molecular dynamics simulation of thermal transport at a nanometer scale constriction in silicon," *Journal of Applied Physics,* vol. 101, p. 074304, 2007.
[61] P. Jund and R. Jullien, "Molecular-dynamics calculation of the thermal conductivity of vitreous silica," *Physical Review B (Condensed Matter),* vol. 59, pp. 13707-11, 1999.
[62] L. Young Hee, R. Biswas, C. M. Soukoulis, C. Z. Wang, C. T. Chan, and K. M. Ho, "Molecular-dynamics simulation of thermal conductivity in amorphous silicon," *Physical Review B (Condensed Matter),* vol. 43, pp. 6573-80, 1991.
[63] S. G. Volz and G. Chen, "Molecular-dynamics simulation of thermal conductivity of silicon crystals," *Physical Review B,* vol. 61, pp. 2651-2656, Jan 2000.
[64] P. Heino, "Simulations of nanoscale thermal conduction," *Microsystem Technologies,* vol. 15, pp. 75-81, 2009.
[65] A. S. Henry and G. Chen, "Spectral phonon transport properties of silicon based on molecular dynamics Simulations and lattice dynamics," *Journal of Computational and Theoretical Nanoscience,* vol. 5, pp. 141-152, Feb 2008.

[66] K. Miyazaki, D. Nagai, Y. Iida, and H. Tsukamoto, "Molecular dynamics simulations of heat conduction in nanostructured silicon," Vancouver, BC, Canada, 2007, pp. 323-328.
[67] F. H. Stillinger and T. A. Weber, "Computer simulation of local order in condensed phases of silicon," *Physical Review B,* vol. 31, p. 5262, 1985.
[68] J. Tersoff, "Empirical interatomic potential for silicon with improved elastic properties," *Physical Review B (Condensed Matter),* vol. 38, pp. 9902-5, 1988.
[69] M. I. Baskes, "Atomistic potentials for the molybdenum-silicon system," *Materials Science and Engineering A,* vol. 261, pp. 165-168, 1999.
[70] B. W. H. van Beest, G. J. Kramer, and R. A. van Santen, "Force fields for silicas and aluminophosphates based on ab initio calculations," *Physical Review Letters,* vol. 64, p. 1955, 1990.
[71] H. Sun, S. J. Mumby, J. R. Maple, and A. T. Hagler, "An ab Initio CFF93 All-Atom Force Field for Polycarbonates," *Journal of the American Chemical Society,* vol. 116, pp. 2978-2987, 2002.
[72] M. Z. Bazant, E. Kaxiras, and J. F. Justo, "Environment-dependent interatomic potential for bulk silicon," *Physical Review B,* vol. 56, p. 8542, 1997.
[73] A. J. Dyson and P. V. Smith, "Extension of the Brenner empirical interatomic potential to C-Si-H systems," *Surface Science,* vol. 355, pp. 140-150, 1996.
[74] A. J. Dyson and P. V. Smith, "Improved empirical interatomic potential for C-Si-H systems," *Molecular Physics,* vol. 96, pp. 1491-507, 1999.
[75] C. Sbraccia, P. L. Silvestrelli, and F. Ancilotto, "Modified XB potential for simulating interactions of organic molecules with Si surfaces," *Surface Science,* vol. 516, pp. 147-158, 2002.
[76] S. MARUYAMA, Y. IGARASHI, Y. TANIGUCHI, and J. SHIOMI, "Anisotropic Heat Transfer of Single-Walled Carbon Nanotubes," *Journal of Thermal Science and Technology,* vol. 1, pp. 138-148, 2006.
[77] J. Shiomi and S. Maruyama, "Diameterand length effect on diffusive-ballistic phonon transport in a carbon nanotube," Vancouver, BC, Canada, 2007, pp. 381-386.
[78] J. Shiomi and S. Maruyama, "Heat conduction of single-walled carbon nanotube isotope superlattice structures: a molecular dynamics study," *Physical Review B (Condensed Matter and Materials Physics),* vol. 74, pp. 155401-1, 2006.
[79] S. Maruyama, "Molecular Dynamics of Diffusive-Ballistic Heat Conduction in Single-Walled Carbon Nanotubes."
[80] S. Maruyama, "A molecular dynamics simulation of heat conduction in finite length SWNTs," Tsukuba, Japan, 2002, pp. 193-195.
[81] S. Maruyama, "A molecular dynamics simulation of heat conduction of a finite length single-walled carbon nanotube," *Microscale Thermophysical Engineering,* vol. 7, pp. 41-50, 2003.
[82] J. Shiomi and S. Maruyama, "Non-Fourier heat conduction in a single-walled carbon nanotube: classical molecular dynamics simulations," *Physical Review B (Condensed Matter and Materials Physics),* vol. 73, pp. 205420-1, 2006.
[83] V. P. Sokhan, D. Nicholson, and N. Quirke, "Phonon spectra in model carbon nanotubes," *Journal of Chemical Physics,* vol. 113, pp. 2007-15, 2000.

[84] M. Alaghemandi, E. Algaer, M. C. Bohm, and F. Muller-Plathe, "The thermal conductivity and thermal rectification of carbon nanotubes studied using reverse non-equilibrium molecular dynamics simulations," *Nanotechnology,* vol. 20, 2009.
[85] J. R. Lukes and H. Zhong, "Thermal conductivity of individual single-wall carbon nanotubes," *Journal of Heat Transfer,* vol. 129, pp. 705-716, 2007.
[86] H. Zhong and J. R. Lukes, "Thermal conductivity of single-wall carbon nanotubes," New York, NY 10016-5990, United States, 2004, pp. 65-73.
[87] M. C. H. Wu and H. Jang-Yu, "Thermal conductivity of carbon nanotubes with quantum correction via heat capacity," *Nanotechnology,* vol. 20, p. 145401 (6 pp.), 2009.
[88] S. Berber, Y.-K. Kwon, and D. Tomanek, "Unusually High Thermal Conductivity of Carbon Nanotubes," *Physical Review Letters,* vol. 84, pp. 4613-4616, 2000.
[89] A. K. Rappe, C. J. Casewit, K. S. Colwell, W. A. Goddard, and W. M. Skiff, "UFF, a full periodic table force field for molecular mechanics and molecular dynamics simulations," *Journal of the American Chemical Society,* vol. 114, pp. 10024-10035, 2002.
[90] D. W. Brenner, O. A. Shenderova, J. A. Harrison, S. J. Stuart, B. Ni, and S. B. Sinnott, "A second-generation reactive empirical bond order (REBO) potential energy expression for hydrocarbons," *Journal of Physics Condensed Matter,* vol. 14, pp. 783-802, 2002.
[91] D. W. Brenner, "Empirical potential for hydrocarbons for use in simulating the chemical vapor deposition of diamond films," *Physical Review B,* vol. 42, p. 9458, 1990.
[92] Y. Yamaguchi and S. Maruyama, "A molecular dynamics simulation of the fullerene formation process," *Chemical Physics Letters,* vol. 286, pp. 336-342, 1998.
[93] J. Tersoff, "Empirical Interatomic Potential for Carbon, with Applications to Amorphous Carbon," *Physical Review Letters,* vol. 61, p. 2879, 1988.
[94] J. Tersoff, "Modeling solid-state chemistry: Interatomic potentials for multicomponent systems," *Physical Review B,* vol. 39, p. 5566, 1989.
[95] J. Tersoff, "New empirical approach for the structure and energy of covalent systems," *Physical Review B (Condensed Matter),* vol. 37, pp. 6991-7000, 1988.
[96] H. Zhong and J. R. Lukes, "Interfacial thermal resistance between carbon nanotubes: Molecular dynamics simulations and analytical thermal modeling (10 pages)," *Physical review. B, Condensed matter and materials physics.,* vol. 74, p. 125403, 2006.
[97] C. F. Carlborg, J. Shiomi, and S. Maruyama, "Thermal boundary resistance between single-walled carbon nanotubes and surrounding matrices," *Physical Review B (Condensed Matter and Materials Physics),* vol. 78, p. 205406 (8 pp.), 2008.
[98] M. Hu, P. Keblinski, J.-S. Wang, and N. Raravikar, "Interfacial thermal conductance between silicon and a vertical carbon nanotube," *Journal of Applied Physics,* vol. 104, 2008.

[99] J. Diao, D. Srivastava, and M. Menon, "Molecular dynamics simulations of carbon nanotube/silicon interfacial thermal conductance," *Journal of Chemical Physics,* vol. 128, Apr 2008.
[100] A. Henry and G. Chen, "High thermal conductivity of single polyethylene chains using molecular dynamics simulations," *Physical Review Letters,* vol. 101, p. 235502, 2008.
[101] J. Che, T. agin, W. Deng, and W. A. Goddard Iii, "Thermal conductivity of diamond and related materials from molecular dynamics simulations," *Journal of Chemical Physics,* vol. 113, pp. 6888-6900, 2000.
[102] T. Kawamura, D. Hori, Y. Kangawa, K. Kakimoto, M. Yoshimura, and Y. Mori, "Thermal conductivity of SiC calculated by molecular dynamics," *Japanese Journal of Applied Physics,* vol. 47, pp. 8898-8901, 2008.
[103] B. Feng, Z. Li, and X. Zhang, "Role of phonon in the thermal and electrical transports in metallic nanofilms," *Journal of Applied Physics,* vol. 105, pp. 104315-7, 2009.
[104] P. Heino and E. Ristolainen, "Thermal conduction at the nanoscale in some metals by MD," *Microelectronics Journal,* vol. 34, pp. 773-7, 2003.
[105] R. N. Salaway, P. E. Hopkins, P. M. Norris, and R. J. Stevens, "Phonon contribution to thermal boundary conductance at metal interfaces using embedded atom method simulations," *International Journal of Thermophysics,* vol. 29, pp. 1987-1996, 2008.
[106] Y. Mishin, M. J. Mehl, D. A. Papaconstantopoulos, A. F. Voter, and J. D. Kress, "Structural stability and lattice defects in copper: Ab initio, tight-binding, and embedded-atom calculations," *Physical Review B (Condensed Matter and Materials Physics),* vol. 63, pp. 224106-1, 2001.
[107] S. Plimpton, "Fast parallel algorithms for short-range molecular dynamics," *Journal of Computational Physics,* vol. 117, pp. 1-19, 1995.
[108] W. G. Hoover, "Canonical dynamics: equilibrium phase-space distributions," *Physical Review A (General Physics),* vol. 31, pp. 1695-7, 1985.
[109] W. G. Hoover, "Constant-pressure equations of motion," *Physical Review A,* vol. 34, p. 2499, 1986.
[110] F. Nishimura, T. Takahashi, K. Watanabe, and T. Yamamoto, "Bending Robustness of Thermal Conductance of Carbon Nanotubes: Nonequilibrium Molecular Dynamics Simulation," *Applied Physics Express,* vol. 2, p. 3, Mar 2009.
[111] P. K. Schelling, S. R. Phillpot, and P. Keblinski, "Comparison of atomic-level simulation methods for computing thermal conductivity," *Physical Review B (Condensed Matter and Materials Physics),* vol. 65, pp. 144306-1, 2002.
[112] A. Maiti, G. D. Mahan, and S. T. Pantelides, "Dynamical simulations of nonequilibrium processes - heat flow and the Kapitza resistance across grain boundaries," *Solid State Communications,* vol. 102, pp. 517-521, 1997.
[113] R. J. Stevens, L. V. Zhigilei, and P. M. Norris, "Effects of temperature and disorder on thermal boundary conductance at solid-solid interfaces: Nonequilibrium molecular dynamics simulations," *International Journal of Heat and Mass Transfer,* vol. 50, pp. 3977-3989, Sep 2007.

[114] P. K. Schelling, S. R. Phillpot, and P. Keblinski, "Kapitza conductance and phonon scattering at grain boundaries by simulation," *Journal of Applied Physics,* vol. 95, pp. 6082-6091, 2004.
[115] Y. F. Chen, D. Y. Li, J. K. Yang, Y. H. Wu, J. R. Lukes, and A. Majumdar, "Molecular dynamics study of the lattice thermal conductivity of Kr/Ar superlattice nanowires," *Physica B-Condensed Matter,* vol. 349, pp. 270-280, Jun 2004.
[116] J. M. Dickey and A. Paskin, "Computer Simulation of the Lattice Dynamics of Solids," *Physical Review,* vol. 188, p. 1407, 1969.
[117] S. Goncalves and H. Bonadeo, "Vibrational densities of states from molecular-dynamics calculations," *Physical Review B (Condensed Matter),* vol. 46, pp. 12019-21, 1992.
[118] W. A. Little, "Kapitza [thermal] resistance between helium and metals in the normal and superconducting states," *Physical Review,* vol. 123, pp. 435-441, 1961.
[119] P. E. Phelan, "Application of diffuse mismatch theory to the prediction of thermal boundary resistance in thin-film high-Tc superconductors," *Transactions of the ASME. Journal of Heat Transfer,* vol. 120, pp. 37-43, 1998.
[120] D. G. Cahill, W. K. Ford, K. E. Goodson, G. D. Mahan, A. Majumdar, H. J. Maris, R. Merlin, and S. R. Phillpot, "Nanoscale thermal transport," *Journal of Applied Physics,* vol. 93, pp. 793-818, 2003.
[121] E. T. Swartz and R. O. Pohl, "Thermal boundary resistance," *Reviews of Modern Physics,* vol. 61, pp. 605-68, 1989.
[122] P. E. Hopkins and P. M. Norris, "Relative Contributions of Inelastic and Elastic Diffuse Phonon Scattering to Thermal Boundary Conductance Across Solid Interfaces," 2009.
[123] P. M. Norris and P. E. Hopkins, "Examining Interfacial Diffuse Phonon Scattering Through Transient Thermoreflectance Measurements of Thermal Boundary Conductance," 2009.
[124] P. E. Hopkins, P. M. Norris, and R. J. Stevens, "Influence of inelastic scattering at metal-dielectric interfaces," *Journal of Heat Transfer,* vol. 130, p. 022401, 2008.
[125] R. J. Stevens, P. M. Norris, and L. V. Zhigilei, "Molecular-dynamics study of thermal boundary resistance: Evidence of strong inelastic scattering transport channels," Anaheim, CA, United states, 2004, pp. 37-46.
[126] M. A. Panzer and K. E. Goodson, "Thermal resistance between low-dimensional nanostructures and semi-infinite media," *Journal of Applied Physics,* vol. 103, p. 094301, 2008.
[127] S. Pettersson and G. D. Mahan, "Theory of the thermal boundary resistance between dissimilar lattices," *Physical Review B,* vol. 42, p. 7386, 1990.
[128] R. J. Stoner and H. J. Maris, "Kapitza conductance and heat flow between solids at temperatures from 50 to 300 K," *Physical Review B (Condensed Matter),* vol. 48, pp. 16373-87, 1993.
[129] D. A. Young and H. J. Maris, "Lattice-dynamical calculation of the Kapitza resistance between fcc lattices," *Physical Review B,* vol. 40, p. 3685, 1989.
[130] H. Zhao and J. B. Freund, "Lattice-dynamical calculation of phonon scattering at ideal Si-Ge interfaces," *Journal of Applied Physics,* vol. 97, pp. 024903-1, 2005.

[131] P. K. Schelling, S. R. Phillpot, and P. Keblinski, "Phonon wave-packet dynamics at semiconductor interfaces by molecular-dynamics simulation," *Applied Physics Letters,* vol. 80, pp. 2484-2486, 2002.
[132] A. Skye and P. K. Schelling, "Thermal resistivity of Si-Ge alloys by molecular-dynamics simulation," *Journal of Applied Physics,* vol. 103, p. 6, Jun 2008.
[133] T. Watanabe, B. Ni, S. R. Phillpot, P. K. Schelling, and P. Keblinski, "Thermal conductance across grain boundaries in diamond from molecular dynamics simulation," *Journal of Applied Physics,* vol. 102, p. 7, Sep 2007.
[134] P. E. Hopkins and P. M. Norris, "Effects of joint vibrational states on thermal boundary conductance," *Nanoscale and Microscale Thermophysical Engineering,* vol. 11, pp. 247-257, Jul-Dec 2007.
[135] P. E. Hopkins, P. M. Norris, R. J. Stevens, T. E. Beechem, and S. Graham, "Influence of interfacial mixing on thermal boundary conductance across a chromium/silicon interface," *Journal of Heat Transfer-Transactions of the Asme,* vol. 130, p. 10, Jun 2008.
[136] T. Beechem, S. Graham, P. Hopkins, and P. Norris, "Role of interface disorder on thermal boundary conductance using a virtual crystal approach," *Applied Physics Letters,* vol. 90, p. 3, Jan 2007.
[137] R. Prasher, T. Tao, and A. Majumdar, "An acoustic and dimensional mismatch model for thermal boundary conductance between a vertical mesoscopic nanowire/nanotube and a bulk substrate," *Journal of Applied Physics,* vol. 102, pp. 104312-1, 2007.
[138] R. Prasher, "Thermal boundary resistance and thermal conductivity of multiwalled carbon nanotubes," *Physical Review B (Condensed Matter and Materials Physics),* vol. 77, pp. 075424-1, 2008.
[139] B. A. Cola, J. Xu, and T. S. Fisher, "Contact mechanics and thermal conductance of carbon nanotube array interfaces," *International Journal of Heat and Mass Transfer,* vol. 52, pp. 3490-3503, 2009.